Exact Sampling of Stationary and Time-Reversed Queues

Jose Blanchet and Aya Wallwater

Columbia University

Abstract

We provide the first algorithm that, under minimal assumptions, allows to simulate the stationary waiting-time sequence of a single-server queue backwards in time, jointly with the input processes of the queue (inter-arrival and service times). The single-server queue is useful in applications of DCFTP (Dominated Coupling From The Past), which is a well known protocol for simulation without bias from steady-state distributions. Our algorithm terminates in finite time assuming only finite mean of the inter-arrival and service times. In order to simulate the single-server queue in stationarity until the first idle period in finite expected termination time we require the existence of finite variance. This requirement is also necessary for such idle time (which is a natural coalescence time in DCFTP applications) to have finite mean. Thus, in this sense, our algorithm is applicable under minimal assumptions.

1 Introduction

It is a pleasure to contribute to this special issue in honor of Professor Don Iglehart, whose scientific contributions have had an enormous impact in the applied probability and stochastic simulation communities. Professor Iglehart research contributions expand areas such as steady-state simulation and queueing analysis. We are glad, in this paper, to contribute to both of these areas from the standpoint of exact (also known as perfect) simulation theory, which aims at sampling without any bias from the steady-state distribution of stochastic systems.

The theory of exact simulation has attracted substantial attention, particularly since the ground breaking paper [13]. In their paper, the authors introduced the most popular sampling protocol for exact simulation to date; namely, Coupling From The Past (CFTP). CFTP is a simulation technique which results in samples from the steady-state distribution of a Markov chain under certain compactness assumptions. The paper [10] describes a useful variation of CFTP, called Dominated CFTP (DCFTP). Like CFTP, DCFTP aims to sample from the steady-state distribution of a Markov chain, but this technique can also be applied to cases in which the state-space is unbounded.

The idea in the DCFTP method is to simulate a dominating stationary process backwards in time until the detection of a so-called coalescence time, in which the target and dominating processes coincide. The sample path of the target process can then be reconstructed forward in time from coalescence up to time zero. The state of the target process at time zero is a sample from the associated stationary distribution.

Our contribution in this paper is to provide, under nearly minimal assumptions (finite-mean service and inter-arrival times), an exact simulation algorithm for the stationary workload of a

single-server queue backwards in time. This is a fundamental queueing system which can be used in many applications as a natural dominating process when applying DCFTP. Usually, additional assumptions, beyond the ones we consider here, have been imposed to enable the simulation of the stationary single-server queue backwards in time. For example, in [14, 15] the author takes advantage of a single-server queue with Poisson arrivals for exact simulation of a multi-server system; see also the recent work of [5], which dramatically improves the running time in [15], but also requires the Poisson arrivals assumption. In the paper [4], under the existence of a finite moment generating function for the service times, the single-server queue, simulated backwards in time, is used to sample from a general class of perpetuities. The paper [3], which builds upon the ideas in [4], also uses the single-server queue backwards in time to sample the state descriptor of the infinite server queue in stationarity; in turn, the infinite-server queue is used to simulate loss networks in stationarity. Other example in which the single-server queue arises as a natural dominating process occurs in the setting of so-called multi-dimensional stochastic-fluid networks, see [2]. Our contribution here allows to extend the applicability these instances, in which the single-server queue has been used as a dominated process under stronger assumptions than the ones we impose here. The extensions are direct in most cases, the multi-server queue with general renewal arrivals requires the application of an additional coupling idea and it is reported in [9].

The first idle period (backwards in time starting from stationarity) is a natural coalescence time when applying DCFTP. Therefore, we are specially interested in an algorithm that has finite expected termination time to simulate such first idle period. Moreover, it is well known that finite-variance service times are necessary if the first idle period (starting from stationarity) has finite expected time (this follows from Wald's identity, [6] p. 178, and from Theorem 2.1 in [1], p. 270). While our algorithm terminates with probability one imposing only the existence of finite mean of service times and inter-arrival times, when we assume finite variances we obtain an algorithm that has finite expected running time (see Theorem 2 in Section 4).

Let us now provide the mathematical description of the problem we want to solve. Consider a random walk $S_n = X_1 + \ldots + X_n$ for $n \geq 1$, and $S_0 = 0$. We assume that $(X_k : k \geq 1)$ is a sequence of independent and identically distributed (IID) random variables with

$$EX_k = 0 \quad \text{and} \quad E\left|X_k\right|^{\beta} < \infty \quad \text{for some } \beta > 1. \tag{1.1}$$

As we indicated earlier, of special interest is the case $E\left|X_k\right|^{\beta} < \infty$ for some $\beta > 2$. Now, for $\mu > 0$ and $n \geq 0$ we define the negative-drift random walk and its associated running (forward) maximum by

$$S_n(\mu) = S_n - n\mu \quad \text{and} \quad M_n = \max_{m \geq n}\{S_m(\mu) - S_n(\mu)\}, \tag{1.2}$$

respectively. Note that the maximum is taken over an infinite time-horizon, so the process $(M_n : n \geq 0)$ is not adapted to the random walk $(S_n(\mu) : n \geq 0)$. Our aim in this paper is to design an algorithm that samples jointly from the sequence $(S_n(\mu), M_n : 0 \leq n \leq N)$ for any finite N (potentially a stopping time adapted to $(S_n(\mu), M_n : n \geq 0)$). Of particular interest is the first idle time, $N = \min\{n \geq 0 : M_n = 0\}$, which can often be used as a coalescence time.

Note that if we define $W_m = M_{-m}$ for $m \leq 0$, then we can easily verify the so-called Lindley's recursion (see [1], p. 92) namely

$$M_{-m} = \left(M_{-m+1} + X_{-m} - \mu\right)^{+} = \left(W_{m-1} + X_{-m} - \mu\right)^{+} = W_m, \tag{1.3}$$

and therefore $(W_m : m \leq 0)$ corresponds to a single-server queue waiting time sequence backwards in time; the sequence is clearly stationary since the M_n's are all equal in distribution. Simulating

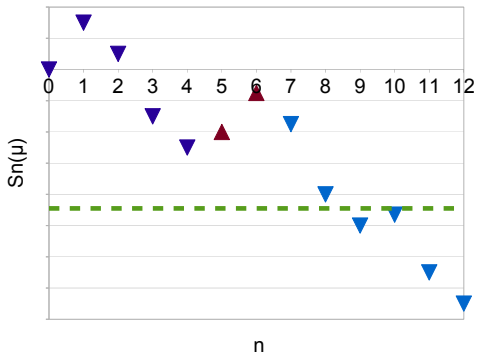


Figure 2.1: **Figure 2.1**
Figure 2.1 illustrates a sample path $\{S_n(\mu) : 0 \leq n \leq 12\}$. If we set $m = 1$ and $L = 2$ then the corresponding stopping times are $D_1 = 4$, $U_1 = 6$, $D_2 = 9$. If in addition $U_2 = \infty$, then $S_n(\mu)$ stays below the bold dashed line for all $n \geq D_2$. Following Proposition 1 we can now evaluate M_n satisfying $\{M_n : n \leq 9, S_n(\mu) \geq S_9(\mu) + 1\}$ In this example, at time $t = D_2 = 9$ the values of $\{M_n : 0 \leq n \leq 7\}$ can be calculated and we can update $C_{UB} \leftarrow S_{D_2}(\mu) + 1$. Notice that $S_8(\mu) \leq S_9(\mu) + 1$ and therefore, in order to determine M_8 we need to keep on tracking the path until the next time we spot $U_n = \infty$.

$(S_n(\mu), M_n : n \geq 0)$ jointly allows to couple the single-server queue backwards in time with the driving sequence (i.e. the X_n's). Such coupling is required in the applications of the DCFTP method.

The algorithm that we propose here extends previous work in [7], which shows how to simulate M_0 assuming the existence of the so-called Cramer root (i.e. $\theta > 0$ such that $E(\exp(\theta X_1)) = 1$). The paper [4] explains how to simulate $(S_n(\mu), M_n : n \geq 0)$ assuming a finite moment generating function in a neighborhood of the origin. Multidimensional extensions, also under the assumption of a finite moment generating function around the origin, are discussed in [2].

Our strategy for simulating the sequence $(S_n(\mu), M_n : n \geq 0)$ relies on certain "upward events" and "downward events" that occur at random times. These "milestone events" will be discussed in Section 2. In Section 2 we will also present the high-level description of our proposed algorithm, which will be elaborated in subsequent sections. Section 3 explains how to simulate M_0 under the assumption that $E|X_k|^\beta < \infty$ for $\beta > 2$. In Section 4 we built on our construction for the sampling of M_0 to simulate the sequence $(S_k(\mu), M_k : k \leq n)$. Section 5 will explain how to extend our algorithm to the case $E|X_k|^\beta < \infty$ for $\beta > 1$ and also discuss additional considerations involved in evaluating certain normalizing constants. Finally, in Section 6 we will present a numerical example that tests the empirical performance of our proposed algorithm.

2 Construction of $(S_n(\mu), M_n : n \geq 0)$ via "milestone events"

We will describe the construction of a pair of sequences of stopping times (with respect to the filtration generated by $(S_n(\mu) : n \geq 0)$), denoted by $(D_n : n \geq 0)$ and $(U_n : n \geq 1)$, which track certain downward and upward milestones in the evolution of $(S_n(\mu) : n \geq 0)$. We follow similar steps as described in [4]. These "milestone events" will be used in the design of our proposed algorithm. The elements of the two stopping times sequences interlace with each other (when finite) and their precise description follows next.

We start by fixing any $m > 0$, $L \geq 1$. Eventually we will choose m as small as possible subject to certain constraints described in Section 3, and then we can choose L as small as possible to

satisfy

$$P\left(m < M_0 \leq (L+1)m\right) > 0. \tag{2.1}$$

Typically, $L = 1$ is feasible. This constraint on L will be used in the proof of Proposition 1 and also in the implementation of Step 2 in Procedure 1 below.

Now set $D_0 = 0$. We observe the evolution of the process $(S_n(\mu) : n \geq 0)$ and detect the time D_1 (the first downward milestone),

$$D_1 = \inf\left\{n \geq D_0 : S_n(\mu) < -Lm\right\}.$$

Once D_1 is detected we check whether or not $\{S_n(\mu) : n \geq D_1\}$ ever goes above the height $S_{D_1}(\mu) + m$ (the first upward milestone); namely we define

$$U_1 = \inf\left\{n \geq D_1 : S_n(\mu) > m + S_{D_1}(\mu)\right\}$$

For now let us assume that we can check if $U_1 = \infty$ or $U_1 < \infty$ (how exactly to do so will be explained in Section 3). To continue simulating the rest of the path, namely $\{S_n(\mu) : n > D_1\}$, we potentially need to keep track of the conditional upper bound implied by the fact that $U_1 = \infty$. To this end, we introduce the conditional upper bound variable C_{UB} (initially $C_{UB} = \infty$). If at time D_1 we detect that $U_1 = \infty$, then we set $C_{UB} = S_{D_1}(\mu) + m$ and continue sampling the path of the random walk conditional on never crossing the upper bound $S_{D_1}(\mu) + m$, that is, conditional on $\{S_n(\mu) < C_{UB} : n > D_1\}$. Otherwise, if $U_1 < \infty$, we simulate the path conditional on $U_1 < \infty$, until we detect the time U_1. We continue on sequentially checking whenever a downward or an upward milestone is crossed as follows: For $j \geq 2$, define

$$\begin{aligned} D_j &= \inf\left\{n \geq U_{j-1} I\left(U_{j-1} < \infty\right) \vee D_{j-1} : S_n(\mu) < S_{D_{j-1}}(\mu) - Lm\right\} \\ U_j &= \inf\left\{n \geq D_j : S_n(\mu) - S_{D_j}(\mu) > m\right\}, \end{aligned} \tag{2.2}$$

with the convention that if $U_{j-1} = \infty$, then $U_{j-1} I\left(U_{j-1} < \infty\right) = 0$. Therefore, we have that $U_{j-1} I\left(U_{j-1} < \infty\right) > D_{j-1}$ if and only if $U_{j-1} < \infty$.

Let us define

$$\Delta = \inf\{D_n : U_n = \infty, n \geq 1\}. \tag{2.3}$$

So, for example, if $U_1 = \infty$ we have that $\Delta = D_1$ and the drifted random walk will never reach level $S_{D_1}(\mu) + m$ again. This allows us to evaluate M_0 by computing

$$M_0 = \max\left\{S_n(\mu) : 0 \leq n \leq \Delta\right\}. \tag{2.4}$$

Similarly, the event $U_j = \infty$, for some $j \geq 1$, implies that the level $S_{D_j}(\mu) + m$ is never crossed for all $n \geq D_j$, and we let $C_{UB} = S_{D_j}(\mu) + m$. The value of C_{UB} keeps updating as the random walk evolves, at times where $U_j = \infty$.

The advantage of considering these stopping times is the following: once we observed that some $U_j = \infty$, the values of $\left\{M_n : n \leq D_j, S_n(\mu) \geq S_{D_j}(\mu) + m\right\}$ are known without a need of further simulation. A detailed example is illustrated in Figure 2.1.

Before we summarize the properties of the stopping times D_n's and U_n's it will be useful to introduce the following. For any a and $b > 0$ let

$$\begin{aligned} T_b &= \inf\left\{n \geq 0 : S_n - \mu n > b\right\}, \\ T_{-b} &= \inf\left\{n \geq 0 : S_n - \mu n < -b\right\}, \\ P_a(\cdot) &= P\left(\cdot \mid S_0 = a\right). \end{aligned} \tag{2.5}$$

Proposition 1. *Set $D_0 = 0$ and let $(D_n : n \geq 1)$ and $(U_n : n \geq 1)$ be as (2.2). We have that*

$$P_0\left(\lim_{n\to\infty} D_n = \infty\right) = 1 \quad \text{and} \quad P_0\left(D_n < \infty\right) = 1, \quad \forall n \geq 1. \tag{2.6}$$

Furthermore,

$$P_0\left(U_n = \infty, \textit{i.o.}\right) = 1. \tag{2.7}$$

Proof. The statement in (2.6) follows easily from the Law of Large Numbers since $ES_1(\mu) = -\mu < 0$. Now we will verify that $P_0(U_n = \infty, \textit{i.o.}) = 1$. Recall that U_1 was defined by $U_1 = \inf\{n \geq D_1 : S_n(\mu) - S_{D_1}(\mu) > m\}$. Therefore, since $ES_1(\mu) < 0$, for all $m \geq 0$ we have (see [1] p. 224),

$$P_0\left(U_1 = \infty | S_1, ..., S_{D_1}\right) = P_0\left(T_m = \infty\right) = P\left(M_0 \leq m\right) \geq P\left(M_0 = 0\right) > 0.$$

Our next goal is to show that for $j \geq 2$ we can find $\delta > 0$ such that

$$P_0\left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) \geq \delta > 0.$$

Suppose first that $U_l < \infty$ for each $l = 1, 2, ..., j-1$. Then, by the strong Markov property we have that

$$P_0\left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) = P_0\left(T_m = \infty\right) \geq P\left(M_0 = 0\right) > 0.$$

Now suppose that $U_l = \infty$ for some $l \leq j-1$ and let $l^* = \max\{l \leq j-1 : U_l = \infty\}$. Define $r = S_{D_{l^*}}(\mu) + m - S_{D_j}(\mu) \geq (L+1)m$. Note that

$$P_0\left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) = P_0\left(T_m = \infty | T_r = \infty\right). \tag{2.8}$$

Keep in mind that the right hand side of (2.8) regards r as a deterministic constant and note that

$$P_0\left(T_m = \infty | T_r = \infty\right) = P_0\left(M_0 \leq m | M_0 \leq r\right) \geq \frac{P_0\left(M_0 = 0\right)}{P\left(M_0 \leq r\right)} \geq P_0\left(M_0 = 0\right) > 0 \tag{2.9}$$

Hence, we conclude that

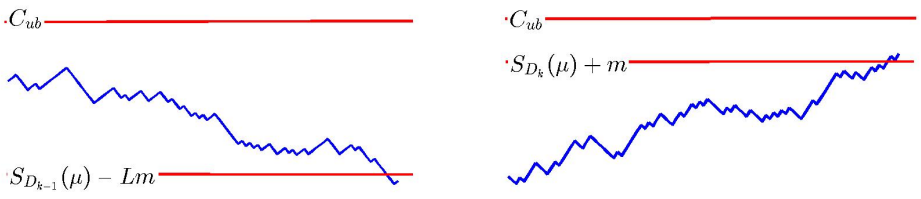
$$P_0\left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) \geq P\left(M_0 = 0\right) := \delta > 0 \ .$$

It then follows by the Borel-Cantelli lemma that $P_0(U_n = \infty, \text{i.o.}) = 1$. □

In the setting of Proposition 1, for each $k \geq 0$ we can define $N_0(k) = \inf\{n \geq 1 : D_n \geq k\}$ and $\mathcal{T}(k) = \inf\{j \geq N_0(k) + 1 : U_j = \infty\}$, both finite random variables such that

$$M_k = -S_k(\mu) + \max\{S_n(\mu) : k \leq n \leq D_{\mathcal{T}(k)}\} \tag{2.10}$$

In words, $D_{\mathcal{T}(k)}$ is the time, not earlier than k, at which we detect a second unsuccessful attempt at building an upward patch directly. The fact that the relation in (2.10) holds, follows easily by construction of the stopping times in (2.2). Note that it is important, however, to define $\mathcal{T}(k) \geq N_0(k) + 1$ so that $D_{N_0(k)+1}$ is computed first. That way we can make sure that the maximum of the sequence $(S_n(\mu) : n \geq k)$ is achieved between k and $D_{\mathcal{T}(k)}$ (see Figure 2.1).



a. Step 1. b. Step 2.

Figure 2.2: High-level description of the algorithm

Proposition 1 ensures that it suffices to sequentially simulate $(D_n : n \geq 0)$ and $(U_n : n \geq 1)$ jointly with the underlying random walk in order to sample from the sequence $(S_n(\mu), M_n : n \geq 0)$. This observation gives rise to our suggested scheme. The procedure sequentially constructs the random walk in the intervals $[D_{n-1}, D_n)$ for $n \geq 1$. Here is the high-level procedure to construct $(S_n(\mu), M_n : n \geq 0)$:

Procedure 1 Milestone Construction of $(S_n(\mu), M_n : n \geq 0)$ (see Figure 2.2)

At kth iteration, $k \geq 1$:

Step 1: "downward patch". Conditional on the path not crossing C_{UB} we simulate the path until we detect $D_k -$ the first time the path crosses the level $S_{D_{k-1}}(\mu) - Lm$ (see Figure 2.2a).

Step 2: "upward patch". Check whether or not the level $S_{D_k}(\mu) + m$ is ever crossed. That is, whether $U_k < \infty$ or not. If the answer is "Yes", then conditional on the path crossing the level $S_{D_k}(\mu) + m$ but not crossing the level C_{UB} we simulate the path until we detect U_k, the first time the level $S_{D_k}(\mu) + m$ is crossed (see Figure 2.2b). Otherwise $(U_j = \infty)$, and we can update C_{UB}: $C_{UB} \leftarrow S_{D_j}(\mu) + m$

The implementation of the steps in Procedure 1 will be discussed in detail in the next sections, culminating with the precise description given in Algorithm 3 at the end of Subsection 4.3. The following result summarizes the main contribution of this paper. The development in the next sections provides the proof of this result, which will ultimately be given after the description of Algorithm 3.

Throughout the rest of the paper a function evaluation is considered to be any of the following operations: evaluation of a sum, a product, the exponential of a number, the underlying increment distribution at a given point, the simulation of a uniform number, and the simulation of a single increment conditioned on lying on a given interval.

Theorem 2. *Suppose that $E|X_k|^\beta < \infty$ for some $\beta > 1$. If $m > 0$ is suitably chosen (see Subsection 3.1.1) then for each $n \geq 0$ deterministic it is possible to simulate exactly the sequence $(D_j : 0 \leq j \leq n)$ and $(U_j : 0 \leq j \leq n)$ jointly with $(S_j(\mu) : j \leq n)$ and therefore (given our previous discussion on the evaluation of M_k), the sequence $(S_k(\mu), M_k : 0 \leq k \leq n)$.*

Moreover, if $\beta > 2$, the expected number of function evaluations required to simulate $(S_k(\mu), M_k : 0 \leq k \leq n)$ is finite. In particular, since $EN < \infty$ for $N = \inf\{k \geq 0 : M_k = 0\}$, the expected running time to simulate $(S_k(\mu), M_k : 0 \leq k \leq N)$ is also finite.

3 Sampling M_0 jointly with $(S_1(\mu), ..., S_\Delta(\mu))$

The goal of this section is to sample exactly from the steady-state distribution of the single-server queue, namely M_0. To this end we need to simulate the sample path up to the first U_j such that $U_j = \infty$ (recall that Δ was defined to be the corresponding D_j). This sample path will be used in the construction of further steps in Procedure 1 .

Throughout this section, in order to simplify the exposition, we will assume that $E|X_k|^{2+\varepsilon} < \infty$ (i.e. $\beta = 2 + \varepsilon$). This will allow us to conclude that our algorithm has finite expected termination time. We will discuss the case $E|X_k|^{1+\varepsilon} < \infty$ only (for $\varepsilon \in (0,1)$) in Section 5 for completeness, but in such case the algorithm may take infinite expected time to terminate.

Let us recall the definition of the crossing stopping times T_b and T_{-b}, for $b > 0$, introduced in (2.5). Since we concentrate on M_0, we have that $C_{UB} = \infty$. We first need to explain a procedure to generate a Bernoulli random variable with success parameter $P_0(T_m < \infty)$, for suitably chosen $m > 0$. Also, this procedure, as we shall see will allow us to simulate $(S_1(\mu), ..., S_{T_m}(\mu))$ given that $T_m < \infty$.

3.1 Sampling $Ber(P_0(T_m < \infty))$ and $(S_1(\mu), ..., S_{T_m}(\mu))$ given $T_m < \infty$

Let us denote by J a Bernoulli random variable with success parameter $P_0(T_m < \infty)$. The constant $m > 0$ will be selected below in Subsection 3.1.1. There are several ways of sampling J, we use a strategy similar to that considered in [12], in connection to a different sampling problem.

In order to sample J we first introduce a partition on the natural numbers (i.e. the positive time line on the lattice) as follows. Let

$$n_k = 2^{k-1}, \quad k = 1, 2, \dots \ . \tag{3.1}$$

This sequence define a partition of the natural numbers via the sets $[n_{k-1}, n_k - 1]$ for $k = 2, 3, ...$ Now, for $k = 2, 3, \dots$ we consider the sets

$$\begin{aligned} A_k &= \bigcup_{j=n_{k-1}}^{n_k-1} \left\{ X_j > (\mu j + m)^{1-\delta} \right\} \\ B_k &= \bigcap_{j=1}^{n_k-1} \left\{ X_j \leq (\mu n_{k-1} + m)^{1-\delta} \right\} \\ & A_k^c \cap B_k^c \end{aligned} \tag{3.2}$$

for some $\delta \in (0, 1/2]$, also to be selected.

First, the algorithm samples the random variable $K \geq 2$, which has probability mass function $g(\cdot)$ that will be specified later. The random variable K relates to the partition on the natural numbers that was induced by (3.1) and $K = k$ will eventually imply that $T_m \in [n_{k-1}, n_k - 1]$. Given $K = k$, the algorithm then proposes a walk $(S_1(\mu), \dots, S_{n_k-1}(\mu))$ via conditioning on one of three possible events described in terms of A_k, $B_k \cap A_k^c$ and $A_k^c \cap B_k^c$ with equal probability (i.e. 1/3 each). Conditioning on A_k and $A_k^c \cap B_k^c$ will be handled using a mixtures based on individual large-jum-events of the form $\{X_j > (\mu j + m)^{1-\delta}\}$. Conditioning on B_k will be handled using an exponential tilting of the distribution of X_j given that $\{X_j < (\mu j + m)^{1-\delta}\}$. The tilting parameter will be selected via

$$\theta_k = \gamma / (n_{k-1}\mu + m), \tag{3.3}$$

for some $\gamma > 0$.

In order to describe all of these conditional sampling procedures we need to provide some definitions and state auxiliary lemmas which will be proved in the appendix.

We will start by specifying the probability mass function $\{g(k), k \geq 2\}$. Consider Y, a Pareto distributed random variable with some regularly varying index $\alpha > 0$, namely,

$$P(Y > y) = \frac{1}{(1+y)^{\alpha}},$$

for $y \geq 0$. Conditions on $\alpha > 0$ will be imposed below. Let

$$\bar{G}(t) = \int_t^{\infty} P(Y > s)\, ds$$

Then we set for $k = 2, 3, \ldots$

$$g(k) = P(K = k) = \frac{\bar{G}(m + \mu n_{k-1}) - \bar{G}(m + \mu n_k)}{\bar{G}(m + \mu n_1)}. \tag{3.4}$$

Let us impose conditions on δ, α, m and γ that will be assumed for the implementation of the algorithm.

3.1.1 Assumptions imposed on the parameters δ, α, m and γ

In addition to $\delta \in (0, 1/2]$, and (2.1), assume that $m \geq 1$ is selected large enough so that

$$\frac{E(X^2)}{m^{2(1-\delta)}} \leq \frac{1}{2}, \tag{3.5}$$

and that the following inequalities hold:

$$\sup_{z \in \mu \cdot \{2^k : k \geq 0\}} \frac{6(1+2z+m)^{\alpha} P\left(X > (z+m)^{1-\delta}\right)}{(\alpha-1)(m+1)^{\alpha-1}\mu} \leq 1, \tag{3.6}$$

$$\sup_{z \in \mu \cdot \{2^k : k \geq 0\}} \frac{\exp\left(-\gamma(m+z)^{\delta} + \frac{\gamma^2 e^{\gamma} E(X^2) z}{(m+z)^{2(1-\delta)}\mu} + 4\frac{z}{\mu} P\left(X > (z+m)^{1-\delta}\right)\right)}{3^{-1}(\alpha-1)(m+1)^{\alpha-1}(1+2z+m)^{-\alpha} z} \leq 1. \tag{3.7}$$

Inequalities (3.6) and (3.7) are used during the proofs of Lemmas 3 and 4 , respectively. Inequality (3.5) appears in a simple technical step leading to (3.7).

In Appendix A we will discuss how equations (3.5)-(3.7) can always be satisfied under our assumptions on the increments X_k.

3.1.2 Some technical lemmas underlying the description of our algorithm

Using the previous assumptions we now are ready to discuss a series of technical lemmas that are the basis for our algorithm.

Lemma 3. *Under (A.2) (see Appendix A) we have that*

$$\frac{3P\left(A_k\right)}{g\left(k\right)} \leq 1,\ \forall k \geq 2. \tag{3.8}$$

Proof. See Appendix B □

On the event B_k we sample the path $\left(S_1\left(\mu\right), ..., S_{n_k-1}\left(\mu\right)\right)$ using an exponential tilting. Specifically, we sample the increments, $\left(X_j : 1 \leq j \leq n_k - 1\right)$, conditional on the event B_k and tilted with parameter θ_k up to time $\min\left\{T_m, n_k - 1\right\}$, where

$$\theta_k = \frac{\gamma}{C_k^{1-\delta}}, \quad \text{and} \quad C_k := \left(n_{k-1}\mu + m\right).$$

Recall that $\gamma > 0$ has been implicitly constrained due to (3.7). The corresponding log-mgf is given by

$$\psi_k\left(\theta_k\right) := \log\left(\frac{E\left[\exp\left\{\theta_k X\right\} I\left(X \leq C_k^{1-\delta}\right)\right]}{P\left(X \leq C_k^{1-\delta}\right)}\right).$$

The likelihood ratio between $P\left(X_j \in \cdot | X_j \leq C_k^{1-\delta}\right)$ and the tilted distribution (to be used in an IID way for $1 \leq j \leq n_k - 1$) denoted via $P_{k,1}\left(\cdot\right)$ is given by

$$\frac{dP_{k,1}}{dP}\left(X\right) = \frac{I\left(X \leq C_k^{1-\delta}\right)\exp\left(\theta_k X - \psi_k\left(\theta_k\right)\right)}{P\left(X \leq C_k^{1-\delta}\right)}. \tag{3.9}$$

Now we summarize some bounds for this likelihood ratio.

Lemma 4. *Under conditions (3.5)-(3.7) we have that*

$$\frac{3\exp(-\theta_k S_{T_m} + T_m \psi_k\left(\theta_k\right))}{g\left(k\right)} \leq 1,\ \forall k \geq 2. \tag{3.10}$$

Proof. See Appendix C □

As the final piece we will note the following.

Lemma 5. *Then, under (A.1), and (A.2) we have that*

$$\frac{3P\left(B_k^c\right)}{g\left(k\right)} \leq 1,\ \forall k \geq 2. \tag{3.11}$$

Proof. See Appendix D □

3.1.3 Algorithm for sampling $Ber\left(P_0\left(T_m < \infty\right)\right)$ jointly with $\left(S_1\left(\mu\right), ..., S_{T_m}\left(\mu\right)\right)$ given $T_m < \infty$

Now we are ready to fully discuss our algorithm to sample J and $\omega = \left(S_1, ...S_{T_m}\right)$ given $T_m < \infty$. In addition to the random variable K following the probability mass function $g\left(\cdot\right)$, let us introduce a random variable Z uniformly distributed on $\{0, 1, 2\}$ and independent of K. Finally, we also introduce $V \sim U\left(0, 1\right)$ independent of everything else.

If $Z = 0$, then we sample the path $\left(S_1, ..., S_{n_k - 1}\right)$ conditional on A_k (denote $P_{k,0}\left(\cdot\right) = P\left(\cdot \middle| A_k\right)$). This will be explained in Subsection 3.1.4, the sample takes $O\left(n_k\right)$ function evaluations to be produced. Then we let

$$J = I(V \leq 3P\left(A_k\right) I(T_m \in [n_{k-1}, n_k - 1])/g\left(k\right)).$$

Owing to Lemma 3, we have that

$$\frac{3P\left(A_k\right)}{g\left(k\right)} \leq 1, \ \forall k \geq 2. \tag{3.12}$$

If $Z = 1$, we sample $\left(S_1\left(\mu\right), ..., S_{n_k - 1}\left(\mu\right)\right)$ by applying each increment X_j conditional on $\{X_j \leq \left(\mu n_{k-1} + m\right)^{1-\delta}\}$ for $j \in \{1, ..., n_k - 1\}$ in an IID way each following the exponential tilting (3.9). This sampling distribution is denoted via $P_{k,1}\left(\cdot\right)$. The simulation of each increment is done using Acceptance/Rejection, as we shall explain, and the overall sampling $\{X_j : j \leq n_k - 1\}$ takes $O\left(n_k\right)$ function evaluations, see Subsections 3.1.5. Additional discussion on the evaluation $\psi_k\left(\theta_k\right)$ in $O\left(n_k\right)$ function evaluations is given in Subsection 5.2. We then set

$$J = I(V \leq 3 \cdot \exp\left\{-\theta_k S_{T_m} + T_m \psi_k\left(\theta_k\right)\right\} I(T_m \in \left[n_{k-1}, n_k - 1\right], A_k^c, B_k)/g\left(k\right)).$$

Observe that Lemma 4 guarantees the inequality

$$\frac{3\exp\left\{-\theta_k S_{T_m} + T_m \psi_k\left(\theta_k\right)\right\}}{g\left(k\right)} \leq 1, \ \forall k \geq 2. \tag{3.13}$$

Finally, if $Z = 2$, we sample the path $\left(S_1\left(\mu\right), ..., S_{n_k - 1}\left(\mu\right)\right)$ conditional on the event B_k^c (denote $P_{k,2}\left(\cdot\right) = P\left(\cdot \middle| B_k^c\right)$). This is done in a completely analogous manner as in Subsection 3.1.4, thus taking $O\left(n_k\right)$ function evaluations. We then let

$$J = I(V \leq 3P\left(B_k^c\right) I(T_m \in \left[n_{k-1}, n_k - 1\right], A_k^c, B_k^c)/g\left(k\right)).$$

Here the inequality

$$\frac{3P\left(B_k^c\right)}{g\left(k\right)} \leq 1, \ \forall k \geq 2, \tag{3.14}$$

is obtained thanks to Lemma 5.

Upon termination we will output the pair (J, ω). If $J = 1$, then we set $\omega = \left(S_1\left(\mu\right), ..., S_{T_m}\left(\mu\right)\right)$. Otherwise $(J = 0)$, we set $\omega = [\,]$, the empty vector. The precise description of the algorithm is given next.

Algorithm 1: Sampling $Ber\left(P_0\left(T_m < \infty\right)\right)$ and $\left(S_1\left(\mu\right), ..., S_{T_m}\left(\mu\right)\right)$ given $T_m < \infty$

Input: $g\left(\cdot\right)$ as in (3.4), with $\alpha, \delta, m, \gamma$ satisfying the conditions in Section 3.1.1 and L as in (2.1).

Output: $J \sim Ber\left(P_0\left(T_m < \infty\right)\right)$ and ω. If $J = 1$, then $\omega = \left(S_1\left(\mu\right), \ldots, S_{T_m}\left(\mu\right)\right)$. Otherwise $(J = 0)$, $\omega = [\,]$ // `If` $J = 0$`, then` ω `equals to the empty vector`

Sample a time K with probability mass function $g\left(k\right) = P\left(K = k\right)$

Sample $Z \sim Unif\left\{0, 1, 2\right\}$

Sample $V \sim U\left(0, 1\right)$ independent of everything

Given Z and $K = k$ sample $\left(S_1, \ldots, S_{n_k}\right)$ as follows:

if $Z = 0$ **then**

  Sample $\tilde{w} = \left(S_j : j \leq n_k - 1\right)$ from $P_{k,0}\left(\cdot\right) := P\left(\cdot \middle| A_k\right)$

  if $V \leq \frac{3P(A_k)}{g(k)} I\left(A_k,\, T_m \in [n_{k-1}, n_k - 1]\right)$ **then**

    $J = 1$

  else

    $J = 0$

if $Z = 1$ **then**

  Sample $\tilde{w} = \left(S_j : j \leq T_m \wedge \left(n_k - 1\right)\right)$ from $P_{k,1}\left(\cdot\right)$

$$dP_{k,1}\left(\tilde{w}\right) = \exp\left\{\theta_k S_{T_m \wedge (n_{k-1})} - \left(T_m \wedge \left(n_k - 1\right)\psi_k\left(\theta_k\right)\right)\right\} dP\left(\tilde{w}\right)$$

  if $V \leq \frac{3\exp\{-\theta_k S_{T_m} + T_m \psi_k(\theta_k)\}}{g(k)} I\left(B_k,\, A_k^c,\, T_m \in [n_{k-1}, n_k - 1]\right)$ **then**

    $J = 1$

  else

    $J = 0$

if $Z = 2$ **then**

  Sample $\tilde{w} = \left(S_j : j \leq n_k - 1\right)$ from $P_{k,2}\left(\cdot\right) := P\left(\cdot \middle| B_k^c\right)$

  if $V \leq \frac{3P(B_k^c)}{g(k)} I\left(B_k^c\, A_k^c,\, T_m \in [n_{k-1}, n_k - 1]\right)$ **then**

    $J = 1$

  else

    $J = 0$

if $J = 1$ **then**

  Output (J, ω), where $\omega = \left(S_j\left(\mu\right) : 1 \leq j \leq T_m\right)$ // `Recall:` $S_j\left(\mu\right) = S_j - \mu j$.

else

  Output (J, ω), where $\omega = [\,]$ and $J = 0$.

We now provide the following result which justifies the validity of the algorithm.

Proposition 6. *The output J is Bernoulli with success parameter $P_0(T_m < \infty)$ and ω follows the required distribution of $(S_1, \ldots, S_{T_m})$ given $T_m < \infty$. Moreover, if $E|X_1|^{2+\varepsilon} < \infty$, then the expected number of function evaluations required to sample J and ω is finite.*

Proof. To verify that indeed $J \sim Ber(P_0(T_m < \infty))$, let $P'(\cdot)$ denote the joint probability distribution of K, Z, $(S_1, ..., S_{n_K - 1})$, and J induced by the algorithm. Note, of course, that $n_K - 1 \geq T_m$ under $P'(\cdot)$. In addition, observe that

$$\begin{aligned} P'(J = 1 | Z = 0, K = k) &= \tfrac{3P(A_k)}{g(k)} \cdot P_0(T_m \in [n_{k-1}, n_k - 1] \,|\, A_k) \\ &= \tfrac{3}{g(k)} \cdot P_0(T_m \in [n_{k-1}, n_k - 1], A_k). \end{aligned} \tag{3.15}$$

Let $r_{k,1} := \exp(-\theta_k S_{T_m} + T_m \psi(\theta_k)) I(B_k, A_k^c, T_m \in [n_{k-1}, n_k - 1])$, and define $E_{k,1}(\cdot)$ to be the expectation operator associated to the exponential tilting distribution with parameter θ_k applied to the random variables $X_1, ..., X_{n_k - 1}$ (see (3.9)). Note that,

$$\begin{aligned} P'(J = 1 | Z = 1, K = k) &= \tfrac{3}{g(k)} E_{k,1}[r_{k,1}] \\ &= \tfrac{3}{g(k)} P_0(B_k, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \end{aligned} \tag{3.16}$$

Finally,

$$P'(J = 1 | Z = 2, K = k) = \tfrac{3}{g(k)} P_0(B_k^c, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \tag{3.17}$$

Combining (3.15)-(3.17) we have

$$\begin{aligned} &P'(J = 1) = \\ &= \sum_{k=2}^{\infty} \tfrac{1}{3} \left(P'(J = 1 | Z = 0, K = k) + P'(J = 1 | Z = 1, K = k) + P'(J = 1 | Z = 2, K = k) \right) g(k) \\ &= \sum_{k=2}^{\infty} \left(P_0(T_m \in [n_{k-1}, n_k - 1], A_k) + P_0(B_k, A_k^c, T_m \in [n_{k-1}, n_k - 1]) + P_0(B_k^c, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \right) \\ &= \sum_{k=2}^{\infty} P_0(T_m \in [n_{k-1}, n_k - 1]) = P_0(T_m < \infty). \end{aligned} \tag{3.18}$$

Similarly we can verify that if $J = 1$, $\omega = (S_1, ..., S_{T_m})$ follows the conditional law $P(\omega \in \cdot | T_m < \infty)$.

Just note that for any F,

$$\begin{aligned} P'(\omega \in F, J = 1 | K = k, Z = 0) &= P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1] \,|A_k) \cdot \tfrac{3P(A_k)}{g(k)} \\ &= P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1], A_k) \cdot \tfrac{3}{g(k)}, \\ P'(\omega \in F, J = 1 | K = k, Z = 1) &= P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1] A_k^c | B_k) \cdot \tfrac{3P(B_k)}{g(k)} \\ &= P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1] A_k^c, B_k) \cdot \tfrac{3}{g(k)}, \\ P'(\omega \in F, J = 1 | K = k, Z = 2) &= P_0(\omega \in F, A_k^c, T_m \in [n_{k-1}, n_k - 1] \,|B_k^c) \cdot \tfrac{3P(B_k^c)}{g(k)} \\ &= P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1], B_k^c, A_k^c) \cdot \tfrac{3}{g(k)}. \end{aligned} \tag{3.19}$$

Consequently, combining these terms

$$\begin{aligned}
&P'(\omega \in F, J=1) \\
&= \sum_{k=2}^{\infty} [P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1], A_k) + P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1] A_k^c, B_k) \\
&+ P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1], B_k^c, A_k^c)] \\
&= \sum_{k=2}^{\infty} P_0(\omega \in F, T_m \in [n_{k-1}, n_k - 1]) = P_0(\omega \in F, T_m < \infty).
\end{aligned} \tag{3.20}$$

Since $P'(J=1) = P_0(T_m < \infty)$, we conclude that indeed

$$P'(\omega \in F | J=1) = P_0(\omega \in F | T_m < \infty).$$

We now argue that the expected number of function evaluations required to generate (J, ω) has finite mean. Let us assume that sampling from $P_{k,0}(\cdot), P_{k,1}(\cdot)$, and $P_{k,2}(\cdot)$ takes $O(n_k)$ function evaluations (a fact that it is not difficult to see, but nonetheless we will justify in Subsections 3.1.4 and 3.1.5). Then, we note that each proposal ω takes on the order of

$$O(\sum_{k=2}^{\infty} n_k g(k)) \leq O(\sum_{k=2}^{\infty} n_k^2 P(Y > n_{k-1}\mu + m)) < \infty$$

function evaluations; the sum is finite assuming that $\alpha > 2$, as indicated in (A.1). □

We close this section explaining how to sample from $P_{k,0}(\cdot), P_{k,1}(\cdot)$, and $P_{k,2}(\cdot)$. We will also verify that it takes $O(n_k)$ function evaluations to sample ω in each of these three cases as claimed in the end of Proposition 6.

3.1.4 Sampling from $P_{k,0}(\cdot)$ and $P_{k,2}(\cdot)$

We now explain how to use Acceptance / Rejection to obtain a sample from $P_{k,0}(\cdot)$ (i.e. sampling $(S_1, ..., S_{n_k-1})$ given A_k). Our proposal distribution, which we denote by $Q(\cdot)$, is based on a mixture $P(\cdot)$ and another distribution which we denote by $\bar{P}(\cdot)$ to be described momentarily. In particular, we shall set $Q = .5P + .5\bar{P}$. As we shall see, the reason for introducing P is to make sure that the acceptance ratio is bounded uniformly over μ. This will be relevant in our discussion on mixing time in heavy-traffic in Section 6 (i.e. when μ is close to zero). If μ is not close to zero then we can simply select $Q = \bar{P}$ and the acceptance ratio will be bounded uniformly in k, but not as $\mu \to 0$.

The distribution of $(S_1, ..., S_{n_k-1})$ under $\bar{P}(\cdot)$ is better described algorithmically. First, we sample T_k with probability mass function $r_k(\cdot)$ given by

$$r_k(j) = \frac{P(X_j > (\mu j + m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})},$$

for $j \in \{n_{k-1}, \ldots, n_k - 1\}$. Next, given $T_k = j$, sample X_j conditional on $X_j > (\mu j + m)^{1-\delta}$. Finally, sample X_i, for $i \neq j$ and $1 \leq i \leq n_k - 1$ from the nominal (unconditional) distribution. We then obtain that

$$\frac{d\bar{P}}{dP}(X_1, ..., X_{n_k-1}) = \frac{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j + m)^{1-\delta}\right)}{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}.$$

Therefore, with $P_{k,0}(\cdot) = P(\cdot|A_k)$ we obtain that

$$\begin{aligned}\frac{I(A_k)}{P(A_k)} \cdot \frac{dP}{dQ}(X_1, ..., X_{n_k-1}) &= 2\frac{I(A_k)}{P(A_k)} \cdot \frac{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j+m)^{1-\delta}\right) + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})} \\ &\leq c_k := \frac{2}{P(A_k)} \cdot \frac{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})}{1+\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})}.\end{aligned} \tag{3.21}$$

Consequently, in order to sample from $P_{k,0}(\cdot)$ it suffices to propose from $Q(\cdot)$ and accept with probability

$$\begin{aligned} q &:= \frac{1}{c_k} \cdot \frac{I(A_k)}{P(A_k)} \cdot \frac{dP}{dQ}(X_1, ..., X_{n_k-1}) \\ &= I(A_k) \cdot \frac{1 + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j+m)^{1-\delta}\right) + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})}. \end{aligned}$$

We note that the expected number of proposals required to accept is c_k. Moreover, as we shall quickly verify, c_k is bounded uniformly both in k and $\mu > 0$. To see this, use the fact that for $x \geq 0$, $1 - x \leq \exp(-x)$ and conclude that

$$\begin{aligned} P(A_k) &= 1 - \prod_{j=n_{k-1}}^{n_k-1} (1 - P(X_j > (\mu j+m)^{1-\delta})) \\ &\geq 1 - \exp\left(-\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})\right). \end{aligned}$$

Let us write

$$\Lambda := \Lambda(k, \mu) = \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j+m)^{1-\delta})$$

and therefore obtain that

$$c_k \leq \frac{2}{1 - \exp(-\Lambda)} \cdot \frac{\Lambda}{1+\Lambda} \leq 4I(\Lambda \in [0, 1/2]) + 6I(\Lambda \geq 1/2) \leq 6.$$

We suggest applying a completely analogous randomization procedure to sample $P_{k,2}(\cdot)$, which corresponds to sampling given the event

$$B_k^c = \bigcup_{j=1}^{n_k-1} \left\{X_j > (\mu n_{k-1} + m)^{1-\delta}\right\}.$$

A very similar argument as the one just discussed shows that the number of proposals required to accept is also uniformly bounded over k and μ. We therefore conclude that it takes $O(n_k)$ function evaluations to sample ω both under $P_{k,0}(\cdot)$ and $P_{k,2}(\cdot)$.

3.1.5 Sampling from $P_{k,1}(\cdot)$

In order to simulate from $P_{k,1}(\cdot)$ we use Acceptance / Rejection. We propose from $P(\cdot)$ (the nominal distribution). Using the fact that $\theta_k = \gamma/C_k^{1-\delta}$, note that

$$\begin{aligned} dP_{k,1} &= \frac{I\left(X \le C_k^{1-\delta}\right)\exp(\theta_k X - \psi_k(\theta_k))}{P\left(X \le C_k^{1-\delta}\right)} dP \\ &\le \frac{I\left(X \le C_k^{1-\delta}\right)\exp(\gamma - \psi_k(\theta_k))}{P\left(X \le C_k^{1-\delta}\right)} dP \le \frac{\exp(\gamma - \psi_k(\theta_k))}{P\left(X \le C_k^{1-\delta}\right)} dP. \end{aligned} \tag{3.22}$$

So, in order to sample from $P_{k,1}(\cdot)$ it suffices to propose from $P(\cdot)$ and accept with probability

$$q(\omega) := \frac{P\left(X \le C_k^{1-\delta}\right)}{\exp(\gamma - \psi_k(\theta_k))} \frac{dP_{k,1}}{dP} = \exp(\theta_k X - \gamma) I\left(X \le C_k^{1-\delta}\right).$$

The expected number of proposals required to obtain a successful sample X from $P_{k,1}(\cdot)$ is equal to

$$\frac{\exp(\gamma - \psi_k(\theta_k))}{P\left(X \le C_k^{1-\delta}\right)} \le \frac{\exp(\gamma)}{P(X \le m)} < \infty,$$

which is clearly uniformly bounded in k. So each increment takes $O(1)$ time to be simulated and therefore we conclude it takes $O(n_k)$ function evaluations to simulate ω under $P_{k,1}(\cdot)$.

3.2 Building M_0 and $(S_1(\mu), ..., S_\Delta(\mu))$ from downward and upward patches

Before we move on to the algorithm let us define the following. Given a vector $\mathbf{s}$, of dimension $d \ge 1$, we let $\mathbf{L}(\mathbf{s}) = \mathbf{s}(d)$ (i.e. the d-th component of the vector $\mathbf{s}$).

Algorithm 2: Sampling M_0 and $(S_1(\mu), ..., S_\Delta(\mu))$

Input: Same as Algorithm 1
Output: The path $(S_1(\mu), ..., S_\Delta(\mu))$
Initialization $\mathbf{s} \leftarrow []$, $F \leftarrow 0$, $\mathbf{L} = 0$
`// initially s is the empty vector,the variable L represents the last position of the drifted random walk`
while $F = 0$ **do**
  Sample $\left(S_1(\mu), \ldots, S_{T_{-Lm}}(\mu)\right)$ given $S_0(\mu) = 0$
  $\mathbf{s} = \left[\mathbf{s}, \mathbf{L} + S_1(\mu), \ldots, \mathbf{L} + S_{T_{-Lm}}(\mu)\right]$
  $\mathbf{L} = \mathbf{L}(\mathbf{s})$
  Call Algorithm 1 and obtain (J, w)
  if $J = 1$ **then**
    Set $\mathbf{s} = [\mathbf{s}, \mathbf{L} + \omega]$
  else
    $F \leftarrow 1$ $(J = 0)$
Output s.

Proposition 7. *The output of Algorithm 2 has the correct distribution according to (2.3) and (2.4). Moreover, if $E|X_1|^{2+\varepsilon} < \infty$, then the expected number of function evaluations required to sample M_0 and $(S_1(\mu), ..., S_\Delta(\mu))$ is finite.*

Proof. The fact that the output has the correct distribution follows directly from our discussion leading to (2.4) and from Proposition 6, which also implies that simulating a single replication of (J, ω) using Algorithm 1 requires finite expected running time. But Algorithm 2 requires a number of calls to Algorithm 1 which is geometrically distributed with mean $1/P_0(T_m = \infty) < \infty$. Therefore, by Wald's identity (see [6], p. 178) we conclude the finite expected running time of Algorithm 2. □

4 From M_0 to $(S_k(\mu), M_k : k \geq 0)$: Implementation of Procedure 1

In this section we will explain in detail how to implement the steps behind the construction of the sequence $(S_n(\mu), M_n : n \geq 0)$ that were described in Procedure 1 . We will be calling Algorithm 1 and Algorithm 2 repeatedly.

4.1 Implementing Step 1 in Procedure 1

In Step 1 we need to sample a downward patch of the drifted random walk $(S_n(\mu) : n \geq 0)$. The goal is to detect the time where the next downward milestone is crossed, namely the next element in the sequence $(D_n : n \geq 1)$, conditional on the event that the level C_{UB} is not crossed. To this end, let us invoke a result in [4].

Lemma 8. *Let $0 < a < b \leq \infty$ and consider any sequence of bounded positive measurable functions $f_k : \mathbb{R}^{k+1} \longrightarrow [0, \infty)$.*

$$E_0\left[f_{T_{-a}}\left(S_0(\mu), ..., S_{T_{-a}}(\mu)\right) | T_b = \infty\right] = \frac{E_0\left[f_{T_{-a}}\left(S_0(\mu), ..., S_{T_{-a}}(\mu)\right) I\left(S_i(\mu) < b,\, \forall i < T_{-a}\right) P_{S_{T_{-a}}}(T_b = \infty)\right]}{P_0(T_b = \infty)}$$

So, if $P^*(\cdot) = P_0(\cdot | T_b = \infty)$, then

$$\frac{dP^*}{dP_0} = \frac{I\left(S_i(\mu) < b, \forall i < T_{-a}\right) P_{S_{T_{-a}}}(T_b = \infty)}{P_0(T_b = \infty)} \leq \frac{1}{P_0(T_b = \infty)}. \tag{4.1}$$

The result of Lemma 8 holds due to the strong Markov property. Lemma 8 enables us to sample a downward patch by means of the Acceptance/Rejection method using the nominal (i.e. unconditional) distribution as proposal. More precisely, suppose that our current position is $S_{D_j}(\mu)$ and we know that the random walk will never reach position C_{UB} (say, if $U_j = \infty$ then $C_{UB} = S_{D_j}(\mu) + m$). Next we need to simulate the path up to time D_{j+1}. Lemma 8 says that we can propose a downward patch $s_1 := S_1(\mu), ..., s_{T_{-Lm}} := S_{T_{-Lm}}(\mu)$, under the nominal probability given $S_0(\mu) = 0$ and $S_i(\mu) \leq m$ for $i \leq T_{-Lm}$. Then we accept the downward patch with probability $P_0(T_\sigma = \infty)$, where $\sigma = C_{UB} - S_{D_j}(\mu) - s_{T_{-Lm}}$. For example, if $U_j = \infty$ then $\sigma = m - s_{T_{-Lm}} \geq (L+1)m$.

Of course, to accept, we can simulate a Bernoulli, say B, with probability $P_0(T_\sigma = \infty)$ by calling Algorithm 1 with $m \longleftarrow \sigma$ and returning $B = 1 - J$. If the downward patch $\left(s_1, ..., s_{T_{-Lm}}\right)$ is accepted we concatenate to produce the output

$$\begin{aligned} & \left(S_0(\mu), ..., S_{D_j}(\mu), S_{D_j+1}(\mu), ..., S_{D_{j+1}}(\mu)\right) \\ = \;& \left(S_0(\mu), ..., S_{D_j}(\mu), S_{D_j}(\mu) + s_1, ..., S_{D_j}(\mu) + s_{T_{-Lm}}\right). \end{aligned}$$

Otherwise, we keep simulating downward-patch proposals until acceptance.

4.2 Implementing Step 2 in Procedure 1

Assume we have finished generating the path up to time D_{j+1} as explained in Subsection 4.1. At this point we let $\sigma = C_{UB} - S_{D_{j+1}}(\mu) \geq (L+1)m$ and define

$$\begin{aligned} \xi &= P_0\left(U_{j+1} < \infty \middle| S_1, \ldots, S_{D_{j+1}}, U_1, \ldots, U_j\right) \\ &= P_0\left(T_m < \infty \middle| T_\sigma = \infty\right) = P_0\left(M_0 > m \,|\, M_0 \leq \sigma\right). \end{aligned}$$

Observe that assumption in equation (2.1) ensures that $\xi > 0$. We will explain how to simulate $B \sim Ber(\xi)$. First, we call Algorithm 2 and obtain the output $\omega = (s_1, ..., s_\Delta)$. We compute M_0 according to (2.4) and keep calling Algorithm 2 until we obtain $M_0 \leq \sigma$, at which point we set $B = I(M_0 > m)$. Of course, we obtain $B \sim Ber(\xi)$ and if $B = 1$ we can write

$$\left(S_{D_{j+1}}(\mu), S_{D_{j+1}+1}(\mu), \ldots, S_{U_{j+1}}(\mu)\right) = \left(S_{D_{j+1}}(\mu), S_{D_{j+1}+1}(\mu) + s_1, ..., S_{D_j}(\mu) + s_\Delta\right). \quad (4.2)$$

Otherwise, $B = 0$, we could simply declare $U_{j+1} = \infty$, update $C_{UB} \leftarrow S_{D_{j+1}}(\mu) + m$ and proceed to the next iteration.

Breaking the path into "upward" and "downward" patches helps to conceptualize the logic of our method. However, it is not an efficient way of implementing the method. A more efficient implementation would be to sequentially generate versions of $\omega = (s_1, ..., s_\Delta)$ as long as $M_0 \leq m$. We can then output the right hand side of (4.2) even when $B = 0$, because the path has been simulated according to the correct distribution given $T_\sigma = \infty$. We provide a precise description of this implementation in Algorithm 3 in the next section.

4.3 Our algorithm to sample $(S_k(\mu), M_k : 0 \leq k \leq n)$ and Proof of Theorem 2

We close this section by giving the explicit implementation of our general method outlined in Subsections 4.1 and 4.2. In order to describe the procedure, let us recall some definitions. Given a vector $\mathbf{s}$ of dimension $d \geq 1$, let $\mathbf{L}(\mathbf{s}) = \mathbf{s}(d)$ (the last element of the vector) and set $\mathbf{d}(\mathbf{s}) = d$ (the length of the vector). The implementation is given in Algorithm 3.

of Theorem 2. The validity of Algorithm 3 is justified following the same logic as in Proposition 7. The only difference here is that the number of trials required to simulate each upward patch is geometrically distributed with a mean which is bounded by $1/P_0(M_0 = 0) < \infty$, following the reasoning behind (2.9). Also note that

$$E_0(T_m I(T_m < \infty)) \leq \sum_{k=2}^{\infty} n_k g(k) < \infty.$$

Moreover, if $\sigma \geq (L+1)m$, by assumption (2.1)

$$E_0\left(T_m | T_m < \infty, T_\sigma = \infty\right) \leq \frac{E_0\left(T_m I(T_m < \infty)\right)}{P_0\left(T_m < \infty, T_\sigma = \infty\right)} \leq \frac{E_0\left(T_m I(T_m < \infty)\right)}{P_0\left(m < M_0 \leq \sigma\right)} < \infty.$$

So, each upward path requires finite number of function evaluations to be produced. The argument for finite expected running time then follows along the lines of Proposition 7. □

Algorithm 3: An Efficient Implementation of Procedure 1

Input: Same as Algorithm 1 and some $n \geq 1$
Output: $(S_k(\mu), M_k : 0 \leq k \leq n)$
Initialization $\mathbf{s} \longleftarrow [0]$, $\mathbf{N} \longleftarrow [0]$, $F \longleftarrow 0$ `// Initialize the sample path with the 1-dimensional zero vector.`
`//` The vector N, which is initially equals to zero records the times D_j such that $U_j = \infty$
`//` F is a Boolean variable which detects when we have enough information to compute M_n
Call Algorithm 2 and obtain $\omega = (s_1, ..., s_\Delta)$
Set $\mathbf{s} = [\mathbf{s}, \omega]$ `// concatenate` ω `to` $\mathbf{s}$
Set $\mathbf{N} = [\mathbf{N}, \mathbf{d}(\mathbf{s}) - 1]$ `// update` $\mathbf{N}$
while $F = 0$ **do**
  if $\mathbf{N}(\mathbf{d}(\mathbf{N}) - 1) \geq n$ **then**
    $F = 1$
  else
    Call Algorithm 2 and obtain $\omega = (s_1, ..., s_\Delta)$, and compute M_0
    if $M_0 \leq m$ **then**
      Set $\mathbf{s} = [\mathbf{s}, \mathbf{L}(\mathbf{s}) + \omega]$
      Set $\mathbf{N} = [\mathbf{N}, \mathbf{d}(\mathbf{s}) - 1]$
for $i = 0, ..., n$ **do**
  $M_i = \max(\mathbf{s}(i+1), \mathbf{s}(i+2), ..., \mathbf{s}(\mathbf{d}(\mathbf{s}))) - \mathbf{s}(i+1)$
  $S_i(\mu) = \mathbf{s}(i+1)$

Output $(S_k(\mu), M_k : 1 \leq k \leq n)$

5 Additional considerations: increments with infinite variance and computing truncated tilted distributions

5.1 Assuming that $E|X|^\beta < \infty$ for $\beta \in (1, 2]$

We will now discuss how to relax the assumption that $E|X|^\beta < \infty$ for $\beta > 2$ and assume only that $E|X|^{1+\varepsilon} < \infty$ for $\varepsilon \in (0, 1]$.

The development can be easily adapted. In order to facilitate the explanation let us discuss the adaptation in the setting of Subsection A, which leads somewhat weaker bounds that those assumed in (3.6) to (3.7) but strong enough to adapt the conclusion in Lemmas 3 to 5.

In order to adapt equation (A.2), for example, we now select $\delta > 0$ small enough so that $1 < \alpha \leq (1+\varepsilon)(1-\delta)$. Then (A.2) is replaced by

$$\frac{6 \cdot 2^\alpha}{(\alpha - 1)(m+1)^{\alpha-1}\mu} \times E\left[\left(X_1^+\right)^{1+\varepsilon}\right] \leq 1.$$

These changes yield that inequality (3.6), which in turn yields the proof Lemma 3 and Lemma 5.

As for Lemma 4, let us now apply Lemma 9 with

$$A(\gamma) = \left(\frac{\gamma^2}{2} \cdot \frac{\exp(1)}{1-\varepsilon} + 2\right) \cdot E\left(|X|^{1+\varepsilon}\right),$$

and obtain

$$\exp(\psi_k(\theta_k)) \leq \exp\left(A(\gamma)\frac{1}{C_k}\right). \tag{5.1}$$

Since T_m we have that $S_{T_m} \geq \mu T_m + m$, and because $T_m \in [n_{k-1}, n_k - 1]$ we conclude that

$$S_{T_m} \geq \mu n_{k-1} + m = C_k.$$

Therefore, on $T_m \in [n_{k-1}, n_k - 1]$

$$\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k)) \leq \exp(-\theta_k C_k + n_k \psi_k(\theta_k)) \leq \exp(-\gamma C_k^\delta + A(\gamma)\frac{n_k}{C_k}) \leq \exp(-\gamma C_k^\delta + 2A(\gamma)/\mu),$$

where the last inequality was obtained from the bound $n_k/C_k \leq n_k/(n_{k-1}\mu)$. So, we conclude, letting $z = \mu n_{k-1}$, that

$$\frac{3\exp(-\gamma C_k^\delta + 2A(\gamma)/\mu)}{g(k)} \leq \frac{3(2z+m)^\alpha}{(\alpha-1)(m+1)^{\alpha-1}z}\exp\left(-\gamma(m+z)^\delta + 2A(\gamma)/\mu\right).$$

Further, if $u = \gamma^{1/\delta}(m+z)$, following the development in Subsection A, we arrive at

$$\begin{aligned}\frac{3\exp(-\gamma C_k^\delta + 2A(\gamma)/\mu)}{g(k)} &\leq \frac{3\cdot 2^\alpha \gamma^{-\alpha/\delta}}{(\alpha-1)(m+1)^{\alpha-1}\mu}\exp(2A(\gamma)/\mu)\max_{u \geq \gamma^{1/\delta} m} u^\alpha \exp\left(-u^\delta\right) \\ &\leq \frac{3\cdot 2^\alpha \gamma^{-\alpha/\delta}}{(\alpha-1)(m+1)^{\alpha-1}\mu}\exp(2A(\gamma)/\mu)\left(\frac{\alpha}{\delta}\right)^\alpha \exp\left(-\left(\frac{\alpha}{\delta}\right)^\delta\right).\end{aligned}$$

For every $\gamma > 0$ we can select m large enough to make the right hand side less than one and this yields the adaptation of the proof of Lemma 4 to the case $\beta \in (1, 2]$.

This discussion implies that Algorithm 3 provides unbiased samples from $(M_k, S_k(\mu) : 0 \leq k \leq n)$ in finite time with probability one. Nevertheless, if $\varepsilon \in (0, 1]$, we have that $\alpha \leq (1-\delta)(1+\varepsilon) < 2$ and therefore the expected number of function evaluations required to sample J in Algorithm 1 is bounded from below by

$$\sum_k n_k^2 P(Y > \mu n_k + m) = \infty.$$

Therefore, the expected running time of Algorithm 3 is not finite.

5.2 The issue of evaluating $\psi_k(\theta_k)$

We are concerned with the evaluation of (3.13), that is, during the course of the algorithm we must decide if

$$V \leq 3 \cdot \exp\{-\theta_k S_{T_m} + T_m \psi_k(\theta_k)\} I(T_m \in [n_{k-1}, n_k - 1], A_k^c, B_k) \tag{5.2}$$

where $V \sim U(0,1)$ independent of S_{T_m} and T_m. In order to decide if inequality (5.2) holds one does not need to compute $\eta_k := \exp(\psi_k(\theta_k))$ explicitly. It suffices to construct a pair of monotone

sequences $\{\eta_k^+(n) : n \geq 0\}$ and $\{\eta_k^-(n) : n \geq 0\}$ such that $\eta_k^+(n) \searrow \eta_k$ as $n \to \infty$ and $\eta_k^-(n) \nearrow \eta_k$ as $n \to \infty$. It is important, however, to have the sequences converging at a suitable speed. For example, it is not difficult to show that if

$$0 \leq \eta_k^+(n) - \eta_k^-(n) \leq c_0 n^{-r}$$

for $r > 2$, and the evaluation of $\eta_k^+(n)$, $\eta_k^-(n)$ takes $O(l(k)n)$ function evaluations then the expected number of function evaluations required to terminate Algorithm 1 will be bounded if $\sum_k g(k) l(k) < \infty$ (this holds if $E|X|^\beta < \infty$ for $\beta > 2$ and $l(k) = O(n_k)$, given our selection of $\alpha > 2$). Note the requirement on quadratic convergence ($r > 2$). Sequences $\eta_k^+(\cdot)$ and $\eta_k^-(\cdot)$ can be constructed assuming the existence of a smooth density for X using quadrature methods. Nevertheless, we do not want to impose the existence of a smooth density and thus we shall advocate a different approach for estimating $\psi_k(\theta_k)$, based on coupling.

The approach that we advocate proceeds as follows. First, note that if X has a lattice distribution, with span $h > 0$, then $\psi_k(\theta_k)$ can be evaluated with $O\left(C_k^{1-\delta}/h\right)$ function evaluations given k. So, the expected number of function evaluations involved in implementing Algorithm 3 and deciding (5.2) is bounded, since $\sum g(k) C_k^{1-\delta} = O(\sum g(k) n_k) < \infty$.

Now, suppose that the distribution of X is non-lattice. The idea is to construct a coupling between $X_j(\mu)$ and a suitably defined lattice-valued random variable $X_j'(\mu')$ so that $X_j(\mu) \leq X_j'(\mu')$, $EX_j' = 0$, and $\mu' > 0$. We will simulate the random walk associated to the $X_j'(\mu')$'s, namely, $S_j'(\mu')$ and the associated sequence $\left(M_j' : j \geq 0\right)$, jointly with $(S_j(\mu) : 0 \leq j \leq n)$. Since $\max\{S_j'(\mu') : j \geq l\} \searrow -\infty$ as $l \to \infty$ we will be able to sample $(M_k : k \leq n)$ after computing N such that $\max\{S_j'(\mu') : j \geq N\} \leq \min\{S_k(\mu) : k \leq n\}$. We now proceed to describe this strategy in detail.

Given $h > 0$ define $X_j' = h\lfloor X_j/h \rfloor - E(h\lfloor X_j/h \rfloor)$; we omit the dependence on h in X_j' for notational convenience. In addition, let $\mu' = \mu - E(h\lfloor X_j/h \rfloor) - h$. Since $E(h\lfloor X_j/h \rfloor) < 0$ for each $h > 0$, if we also select $h \leq \mu$ we have $\mu' > 0$. Define

$$X_j'(\mu') = X_j' - \mu' = h\lfloor X_j/h \rfloor - \mu + h,$$

and note that

$$X_j'(\mu') \geq X_j(\mu).$$

We then define the corresponding random walks $S_n' = X_1' + ... + X_n'$, $S_n'(\mu') = S_n' - n\mu'$ with $S_0' = 0$ and

$$M_n'(\mu') = \sup\{S_k'(\mu') : k \geq n\} - S_n'(\mu').$$

The following algorithm summarizes our strategy to simulate $(S_k(\mu), M_k : 0 \leq k \leq n)$ when $\psi_k(\theta_k)$ cannot be computed exactly.

The complexity analysis (i.e. finite expected running time if $E|X_1|^{2+\varepsilon} < \infty$) carries over since $EM_0' < \infty$, $E|\min\{S_k(\mu) : k \leq n\}| < \infty$, and therefore $EN < \infty$, with N defined in Algorithm 4.

6 Numerical Example

We will now illustrate our algorithm by revisiting the example that was described in the Introduction. This example considers the waiting time sequence that corresponds to the single-server queue. Recall that this sequence $(W_n : n \geq 0)$ can be generated by the so-called Lindley's recursion

$$W_n = (W_{n-1} + X_n - \mu)^+ \tag{6.1}$$

Algorithm 4: Strategy for simulating $(S_k(\mu), M_k : 0 \le k \le n)$

Input: Same as Algorithm 1 but for X'_j and $h \in (0, \mu)$
Output: $(S_k(\mu), M_k : 1 \le k \le n)$
Call Algorithm 3 and obtain $\omega' = (S'_k(\mu'), M'_k : 0 \le k \le n)$
Given $\omega' = (S'_k(\mu') : 0 \le k \le n)$ sample $\omega = (S_k : 0 \le k \le n)$; `// this is done by sampling` X_k `given the simulated outcome of` $\lfloor X_k/h \rfloor$
Set $M_n^- := \min(S_k(\mu) : 0 \le k \le n)$
Using Algorithm 3, continue sampling $(S'_k(\mu'), M'_k : n \le k \le N)$, where $N = \inf\{k \ge n : M'_k + S'_k(\mu') \le M_n^-\}$
Given $(S'_k(\mu') : n \le k \le N)$, sample $(S_k : n \le k \le N)$
Set $M_k = \max\{S_j(\mu) : k \le j \le N\} - S_k(\mu)$ for $0 \le k \le n$
Output $(S_k(\mu), M_k : 0 \le k \le n)$.

and when in steady state, the W_n's are equal in distribution to

$$M_0 = \max_{m \ge 0}\{S_m(\mu)\}$$

To demonstrate the capability of our algorithm, we chose a sequence of X_n's of the form

$$X_n = h\left\lfloor \frac{c}{h} V_n \right\rfloor - E\left(h\left\lfloor \frac{c}{h} V_n \right\rfloor\right) =: Y_n - E(Y_n) \tag{6.2}$$

where $V_n \sim Pareto(\alpha')$, that is,

$$P(V > t) = \frac{1}{(1+t)^{\alpha'}} \qquad t > 0$$

The parameters α', c, and h can be changed in order to test the algorithm in different scenarios. $\alpha' > 2$ determines how heavy the tail of the increments is, $h > 0$ is the lattice parameter (the non-lattice case is where $h \to 0$), and $c > 0$ controls the mean of Y_n.

6.1 Choice of Parameters

As mentioned at the end of Subsection A, we used the Excel solver in the following way: given our selection of $\alpha \in (2, 4)$, we picked $\delta \in (0, 1/2]$, $\gamma \ge 0$, and $m \ge 0$ so as to minimize m subject to (3.6) and (3.7). The input parameters μ, α', h, and c are chosen to test conditions ranging from light to heavy traffic (controlled primarily by the parameter μ), and from heavy tails to relatively lighter tails (which are controlled by the parameter α').

We conclude our discussion by providing a formal comparison against the relaxation time of the Markov chain $\{W_n : n \ge 0\}$ in heavy-traffic. We chose a formal comparison because a rigorous computation of the exact relaxation time of the single-server queue is not available (to the best of our knowledge) at the level of generality at which our algorithm works, although bounds have been studied, as is the case in [8]. We have argued that our algorithm is sharp, in the sense that it is applicable under close to minimal conditions required for the stability of the single-server queue. We believe that the heavy-traffic analysis provides yet another interesting perspective.

Assuming that $\beta > 2$ (i.e. the increments have finite variance), in heavy traffic, as $\mu \to 0$, it is well known that at temporal scales of order $O\left(1/\mu^2\right)$ and spatial scales of order $O(1/\mu)$ Lindley's

recursion can be approximated by a one dimensional reflected Brownian motion (RBM). In fact, the approximation persists also for the corresponding stationary distribution (which converges after proper normalization to an exponential distribution, which is the stationary distribution of RBM (see [11], for example)). The relaxation time of $\{W_n : n \geq 0\}$ is of order $O\left(1/\mu^2\right)$ as $\mu \to 0$.

The running time analysis of our algorithm involves the "downward" patches, which take $O(m)$ random numbers to be produced. We also need to account for the simulation of the Bernoulli trials for each "upward" patch, which requires the generation of K under $g(\cdot)$, and a total of $C_0 = O(\sum_{k=1}^{\infty} n_k g(k))$ expected random numbers to be simulated. This analysis holds because the number of proposals required to sample $P_{k,0}$, $P_{k,1}$ and $P_{k,2}$ remains bounded also as $\mu \to 0$. Therefore, the actual X_i's conditional on the E_i's can be easily simulated. A similar strategy can be implemented for $P_{k,2}$.

Consequently, the over all cost of our algorithm is driven by $C_0 = O\left(\mu^{-2}m\right)$. We also need to ensure that m is selected so that (3.6) and (3.7) are satisfied. From the analysis of (A.2) and (A.4), we see that $m = O\left(\mu^{-1}\right)$ is always a possible choice. However, this choice can be improved if one can select a large α , which in turn is feasible as long as $z^{\alpha}P\left(X > z^{1-\delta}\right) = O(1)$. In particular, we can choose $m = O\left(1/\mu^{1/(\alpha-1)}\right)$, provided that δ is chosen sufficiently close to unity in order to satisfy (A.4). Our exact sampling algorithm in heavy traffic has a running time that is not worse that $O\left(1/\mu^3\right)$ and it can be arbitrarily close to the relaxation time $O\left(1/\mu^2\right)$ of the chain $\{W_n : n \geq 0\}$.

6.2 Simulation Results

We tested the algorithm in four different cases in which we changed the nature of the random walk increments and the traffic intensity. By picking $\alpha' = 2.9$ and $\alpha' = 7$, we considered heavy tailed increments and relatively lighter tailed increments, respectively. By changing the value of c, we changed the traffic intensity ρ, which is given by

$$\rho = \frac{E\left(h\left\lfloor \frac{c}{h}V \right\rfloor\right)}{E\left(h\left\lfloor \frac{c}{h}V \right\rfloor\right) + \mu} \approx \frac{cE(V)}{cE(V) + \mu}$$

Throughout all scenarios we used the parameters

$$L = 1.1, \quad h = 0.1, \quad \mu = 1 \quad and \quad \delta = 0.38$$

The rest of the parameters were chosen as follows:

	$\rho = 0.3$				$\rho = 0.8$			
	α	γ	c	m	α	γ	c	m
$\alpha' = 7$	4	1.7	3	16	4	0.75	25	217
$\alpha' = 2.9$	2.01	1.24	0.85	35	2.01	0.74	8	400

In each of the above cases we generated 100,000 exact replicas of M_0 and compared it with the chain $\{W_n : 0 \leq n \leq l\}$, where l was picked to fit the scenario. To analyze the output of the chain, we used batches with varying sizes. In the light traffic case, for both $\alpha' = 2.9$ and $\alpha' = 7$, we used $l = 10^6$ with batches of size 25. In the heavy traffic scenario, we used $l = 2 \cdot 10^6$ with batches of size 50 for $\alpha' = 7$, and $l = 4 \cdot 10^6$ with batches of size 100 for $\alpha' = 2.9$
We summarized the result in the following table (see also Figure 6.1):

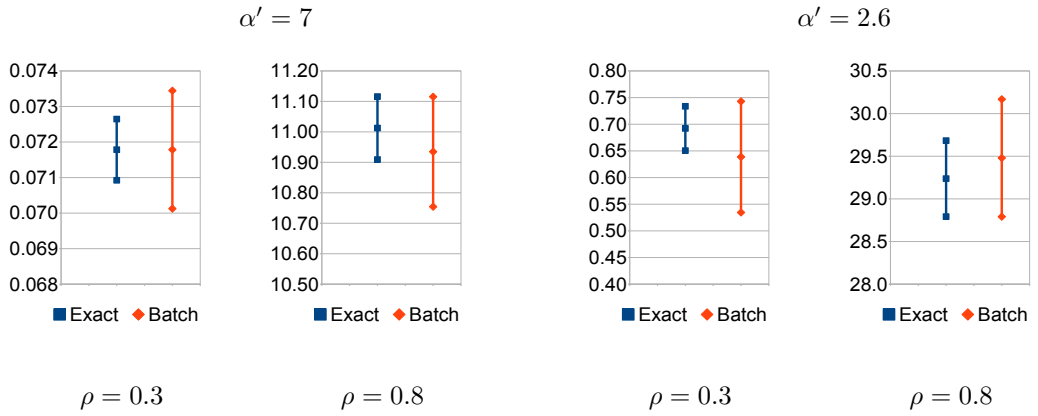


Figure 6.1: Exact sampler mean $E(M_0)$ VS. batches mean of $\{W_n : 0 \leq n \leq l\}$, along with the corresponding 95% confidence intervals.

		$\rho = 0.3$			$\rho = 0.8$		
		LCI	UCI	RT	LCI	UCI	RT
$\alpha' = 7$	Exact sampler	0.0709	0.0726	≈ 1.5	10.9092	11.1159	≈ 10
	Batch mean	0.0701	0.0734	≈ 1	10.7542	11.1152	≈ 3
$\alpha' = 2.9$	Exact sampler	0.6505	0.7336	≈ 3	28.7925	29.6832	≈ 15
	Batch mean	0.5344	0.7429	≈ 1	28.7908	30.1681	≈ 4

LCI/UCI=Lower/Upper 95% Confidence Interval. RT= Running Time (in minutes).

In the numerical examples we see that the IID replications of M_0 appear to be a reasonable approach to steady-state estimation, especially in light traffic. The performance deteriorates somewhat in heavy traffic, which is expected given our earlier discussion on running time in heavy traffic. Nevertheless, it is important to note that while our procedure does not have any bias, batch means do not provide control on the bias with absolute certainty. Overall, we feel that a few minutes of additional running time in exchange for total bias deletion is not an onerous price to pay. Therefore, our procedure is not only of theoretical interest (as the first exact sampler for a general single-server queue), but of practical value as well.

7 Conclusions

The work presented in this paper was motivated by the important role that single-server queue plays in many applications that use the DCFTP method as well as the challenge of efficiently dealing with random walks involving heavy-tailed increments. We developed an exact simulation method that can be used to simulate the stationary waiting-time sequence of a single-server queue backward

in time, jointly with the input process of the queue. We provided an algorithm, which is easy to implement, that has a finite expected termination time under nearly minimal assumptions.

APPENDIX

A Discussion on the generality of the assumptions imposed and selection of parameters

In this section we will argue that the inequalities (3.5)-(3.7) can always be satisfied under our underlying assumption that $E|X_k|^\beta < \infty$ for $\beta = 2+\varepsilon > 2$ (the case $\beta > 1$ is discussed in Section 5). First, the selection of L in (2.1) is always feasible, as indicated earlier $L = 1$ is most of the time feasible; for example $L = 1$ will be feasible if X_1 is non-lattice.

Clearly the selection of m satisfying (3.5) is always feasible. Now, note that we can always select $\delta > 0$ so that

$$2 < \alpha \leq (2+\varepsilon)(1-\delta). \tag{A.1}$$

Then observe that if $m \geq 1$, applying Chebyshev's inequality,

$$\begin{aligned} & \frac{6(1+2\mu z+m)^\alpha}{(\alpha-1)(m+1)^{\alpha-1}\mu} P(X > (\mu z+m)^{1-\delta}) \\ \leq \quad & \frac{6\cdot 2^\alpha(\mu z+m)^\alpha}{(\alpha-1)(m+1)^{\alpha-1}\mu} \times \frac{E\left[\left(X_1^+\right)^{2+\varepsilon}\right]}{(\mu z+m)^{(2+\varepsilon)(1-\delta)}} \leq \frac{6\cdot 2^\alpha}{(\alpha-1)(m+1)^{\alpha-1}\mu} \times E\left[\left(X_1^+\right)^{2+\varepsilon}\right]. \end{aligned}$$

So, condition (3.6) is automatically satisfied if m is chosen sufficiently large so that

$$\frac{6\cdot 2^\alpha}{(\alpha-1)(m+1)^{\alpha-1}\mu} \times E\left[\left(X_1^+\right)^{2+\varepsilon}\right] \leq 1. \tag{A.2}$$

Next, for (3.7), we optimize over z and obtain

$$\frac{z}{(m+z)^{2(1-\delta)}} \leq \frac{1}{m^{1-2\delta}} \cdot \frac{(1-2\delta)^{1-2\delta}}{(2(1-\delta))^{2(1-\delta)}}, \tag{A.3}$$

for all $\delta \in (0, 1/2]$. Use Chebyshev's inequality, together with (A.3), and the change of variable $u = \gamma^{1/\delta}(m+z)$ to obtain

$$\begin{aligned} & \frac{3(1+2z+m)^\alpha}{(\alpha-1)(m+1)^{\alpha-1}z} \exp\left(-\gamma(m+z)^\delta + \frac{\gamma^2 e^\gamma E\left(X^2\right)z}{(m+z)^{2(1-\delta)}\mu} + 4\frac{z}{\mu}P\left(X > (z+m)^{1-\delta}\right)\right) \\ \leq \quad & \frac{3(1+2z+m)^\alpha}{(\alpha-1)(m+1)^{\alpha-1}z} \exp\left(-\gamma(m+z)^\delta + \frac{(\gamma^2 e^\gamma+4)E\left(X^2\right)(1-2\delta)^{1-2\delta}}{(2(1-\delta))^{2(1-\delta)}\mu m^{1-2\delta}}\right). \\ \leq \quad & \frac{3\cdot 2^\alpha \gamma^{-\alpha/\delta}}{(\alpha-1)(m+1)^{\alpha-1}\mu} \exp\left(\frac{(\gamma^2 e^\gamma+4)E\left(X^2\right)(1-2\delta)^{1-2\delta}}{(2(1-\delta))^{2(1-\delta)}\mu m^{1-2\delta}}\right) \max_{u \geq \gamma^{1/\delta} m} u^\alpha \exp\left(-u^\delta\right). \end{aligned}$$

Thus, we can first select $\gamma = 1$, for example, and then pick the smallest m so that

$$\frac{3 \cdot 2^{\alpha}}{(\alpha-1)(m+1)^{\alpha-1}\mu} \exp\left(\frac{7E\left(X^2\right)(1-2\delta)^{1-2\delta}}{(2(1-\delta))^{2(1-\delta)}\mu m^{1-2\delta}}\right) \max_{u \geq \gamma^{1/\delta} m} u^{\alpha} \exp\left(-u^{\delta}\right) \leq 1. \tag{A.4}$$

This can be done numerically or, explicitly by simply by noting (using elementary calculus) that

$$\max_{u \geq \gamma^{1/\delta} m} u^{\alpha} \exp\left(-u^{\delta}\right) \leq \left(\frac{\alpha}{\delta}\right)^{\alpha} \exp\left(-\left(\frac{\alpha}{\delta}\right)^{\delta}\right).$$

In the numerical examples that we will discuss in Section 6 we noted that the performance of the algorithm is not too sensitive to the selection of α, and thus we advocate picking α somewhat larger than 2, for instance $\alpha \in (2, 4]$, but it is important to constrain α and δ so that $z^{\alpha} P\left(X > z^{1-\delta}\right) = O(1)$, due to (3.6).

It is constraint (3.7) the one that has the highest impact in the algorithm's performance and we noted that the selection of m, in particular, was the most relevant parameter. So, we simply used the Excel solver; given our selection of α we picked $\delta \in (0, 1/2]$, $\gamma \geq 0$ and $m \geq 0$ so as to minimize m subject to (3.6) and (3.7). The optimization is done only once and it took a second.

In Section 6 we will also argue that the running time of our algorithm is close to the relaxation time of the Markov chain from a heavy-traffic perspective.

B Proof of Lemma 3

Proof. Notice that

$$\begin{aligned} P(A_k) &\leq \sum_{j=n_{k-1}}^{n_k-1} P\left(X_j > (j\mu + m)^{1-\delta}\right) \\ &\leq n_k P\left(X_1 > (n_{k-1}\mu + m)^{1-\delta}\right). \end{aligned}$$

It is straightforward to verify (using Chebyshev's inequality, the fact that $E|X_1|^{\beta} < \infty$ for $\beta > 1$ and the definition of n_k) that for any $\delta > 0$,

$$\sum_k n_k P\left(X_1 > (n_{k-1}\mu + m)^{1-\delta}\right) < \infty.$$

Now we have for $k \geq 2$

$$\begin{aligned} \tfrac{3P(A_k)}{g(k)} &\leq 3\bar{G}(m) \tfrac{n_k P(X_1 > (n_{k-1}\mu+m)^{1-\delta})}{\int_{m+\mu n_{k-1}}^{m+\mu n_k} P(Y>s)ds} \\ &\leq 3\bar{G}(m) \tfrac{n_k P(X_1 > (\mu n_{k-1}+m)^{1-\delta})}{\mu n_{k-1} P(Y > m+n_k)} \\ &= 6\bar{G}(m) \tfrac{P(X_1^+ > (\mu n_{k-1}+m)^{1-\delta})}{\mu P(Y > m+\mu n_k)} \leq \tfrac{6(1+2\mu n_{k-1}+m)^{\alpha}}{(\alpha-1)(m+1)^{\alpha-1}\mu} P\left(X > (\mu n_{k-1}+m)^{1-\delta}\right) \leq 1 \end{aligned} \tag{B.1}$$

Making $z = \mu n_{k-1} = \mu 2^{k-2}$ and using (3.6) we obtain the conclusion of the lemma. □

C Proof of Lemma 4

Before we prove Lemma 4, we will first introduce an auxiliary lemma, which will be proved at the end of this section.

Lemma 9. *Set* $\theta = \gamma/u^{1-\delta}$ for $\delta \in (0,1)$, $u, \gamma > 0$ *and* suppose that $E(X) = 0$. *If* $E\left(|X|^{1+\varepsilon}\right) < \infty$ *for some* $\varepsilon \in (0,1)$ *and*

$$\frac{E\left(|X|^{1+\varepsilon}\right)}{u^{(1-\delta)(1+\varepsilon)}} \leq \frac{1}{2}, \tag{C.1}$$

then

$$E\left[\exp(\theta X) \mid X \leq u^{1-\delta}\right] \leq \exp\left\{\frac{A}{u^{(1-\delta)(1+\varepsilon)}}\right\} \tag{C.2}$$

with

$$A = \left(\frac{\gamma^2}{2} \cdot \frac{\exp(\gamma)}{1-\varepsilon} + 2\right) \cdot E\left(|X|^{1+\varepsilon}\right). \tag{C.3}$$

Moreover, if $E\left(X^2\right) < \infty$ *and*

$$\frac{E\left(X^2\right)}{u^{2(1-\delta)}} \leq \frac{1}{2} \tag{C.4}$$

then

$$E\left[\exp(\theta X) \mid X \leq u^{1-\delta}\right] \leq \exp\left(\frac{\gamma^2 \exp(\gamma) E(X^2)}{2u^{2(1-\delta)}} + 2P\left(X > u^{1-\delta}\right)\right) \leq \exp\left\{\frac{A}{u^{2(1-\delta)}}\right\}, \tag{C.5}$$

with

$$A = \left(\frac{\gamma^2 \exp(\gamma)}{2} + 2\right) \cdot E\left(X^2\right). \tag{C.6}$$

If in addition $u \geq 1$ *and* $0 < \delta \leq \varepsilon/2$ *then from (C.2) we obtain*

$$E\left[\exp(\theta X) \mid X \leq u^{1-\delta}\right] \leq \exp\left(\frac{A}{u}\right), \tag{C.7}$$

and if $EX^2 < \infty$ *inequality (C.7) follows from (C.5) choosing* $0 \leq \delta \leq 1/2$.

Having Lemma 9 at hand we are now ready to prove Lemma 4

Proof of Lemma 4. Since $m \geq 1$ satisfies inequality (3.5), then we can invoke Lemma 9 with $u = n_{k-1}\mu + m = C_k$ and obtain

$$\exp(\psi_k(\theta_k)) \leq \exp\left(\frac{\gamma^2 \exp(\gamma) E(X^2)}{2C_k^{2(1-\delta)}} + 2P\left(X > C_k^{1-\delta}\right)\right). \tag{C.8}$$

By definition of T_m we have that $S_{T_m} \geq \mu T_m + m$, and because $T_m \in [n_{k-1}, n_k - 1]$ we conclude that

$$S_{T_m} \geq \mu n_{k-1} + m = C_k.$$

Therefore, on $T_m \in [n_{k-1}, n_k - 1]$

$$\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k)) \leq \exp(-\theta_k C_k + n_k \psi_k(\theta_k)). \tag{C.9}$$

Combining (C.8) and (C.9), and letting $z = \mu n_{k-1}$, we obtain that

$$\begin{aligned} & \exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k)) \\ \leq & \exp\left(-\gamma(\mu n_{k-1} + m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) n_{k-1}}{(\mu n_{k-1} + m)^{2(1-\delta)}} + 2n_k P\left(X > (\mu n_{k-1} + m)^{(1-\delta)}\right)\right) \\ = & \exp\left(-\gamma(z+m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) z}{(z+m)^{2(1-\delta)} \mu} + 4\frac{z}{\mu} P\left(X > (z+m)^{(1-\delta)}\right)\right). \end{aligned}$$

So, using (3.7) we conclude that

$$\begin{aligned} & \frac{3\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k))}{g(k)} \\ \leq & \frac{3(1+2z+m)^{\alpha}}{(\alpha-1)(m+1)^{\alpha-1} z} \exp\left(-\gamma(z+m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) z}{(z+m)^{2(1-\delta)} \mu} + 4\frac{z}{\mu} P\left(X > (z+m)^{(1-\delta)}\right)\right) \leq 1, \end{aligned}$$

thereby obtaining the result. □

We conclude this appendix with the proof of the auxiliary lemma.

Proof of Lemma 9. Since $EX = 0$, $E[XI\left(X \leq u^{1-\delta}\right)] < 0$, and therefore a Taylor expansion of second order yields

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq 1 + \frac{\gamma^2}{2} E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 \exp\left\{\frac{\gamma X}{u^{1-\delta}}\right\} I\left(X \leq u^{1-\delta}\right)\right]$$

If $EX^2 < \infty$, we conclude that

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq 1 + \frac{\gamma^2 \exp(\gamma)}{2} \cdot E(X^2) \cdot \frac{1}{u^{2(1-\delta)}}.$$

Since $1 + x \leq \exp(x)$ for $x \geq 0$ we conclude that

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq \exp\left(\frac{\gamma^2 \exp(\gamma)}{2} \cdot E(X^2) \cdot \frac{1}{u^{2(1-\delta)}}\right).$$

On the other hand

$$P\left(X \leq u^{1-\delta}\right) = 1 - P\left(X > u^{1-\delta}\right) \geq 1 - \frac{E(X^2)}{u^{2(1-\delta)}}$$

and since $1 - x \geq \exp(-2x)$ for $x \in (0, 1/2)$ we conclude that if (C.4) holds then

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\} \mid X \leq u^{1-\delta}\right] \leq \exp\left(\frac{\gamma^2 \exp(\gamma) E(X^2)}{2u^{2(1-\delta)}} + 2P\left(X > u^{1-\delta}\right)\right),$$

which yields (C.5). Now, let's assume that $\varepsilon \in (0,1)$ and $E|X|^{1+\varepsilon} < \infty$. Since $z^2 \exp(-z) \leq 4\exp(-2) < 1$ for $z \geq 0$ we have that

$$E\left[\left(\frac{X\gamma}{u^{1-\delta}}\right)^2 \exp\left\{\frac{X\gamma}{u^{1-\delta}}\right\} I\left(X \leq u^{1-\delta}\right)\right] \leq \gamma^2 \exp(\gamma) E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 I\left(|X| \leq u^{1-\delta}\right)\right] + P\left(X < -u^{1-\delta}\right).$$

In addition,

$$E\left[|X|^2 I\left(|X| \leq u^{1-\delta}\right)\right] = 2E\left[\int_0^{u^{1-\delta}} sI\left(|X| > s\right) ds\right] = 2\int_0^{u^{1-\delta}} sP\left(|X| > s\right) ds \leq \frac{E|X|^{1+\varepsilon}}{1-\varepsilon} u^{(1-\varepsilon)(1-\delta)}$$

Therefore,

$$E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 I\left(|X| \leq u^{1-\delta}\right)\right] \leq \frac{E|X|^{1+\varepsilon}}{1-\varepsilon} \cdot \frac{1}{u^{(1+\varepsilon)(1-\delta)}}.$$

Since

$$P\left(X < -u^{1-\delta}\right) \leq \frac{E|X|^{1+\varepsilon}}{u^{(1+\varepsilon)(1-\delta)}},$$

we conclude combining (C) and (C) that

$$\begin{aligned} E\left[\exp\left\{X\tfrac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] &\leq 1 + \tfrac{\gamma^2}{2} \cdot E|X|^{1+\varepsilon} \cdot \left(\tfrac{\exp(\gamma)}{(1-\varepsilon)} + 1\right) \cdot \tfrac{1}{u^{(1+\varepsilon)(1-\delta)}} \\ &\leq 1 + \gamma^2 \cdot E|X|^{1+\varepsilon} \cdot \tfrac{\exp(\gamma)}{(1-\varepsilon)} \cdot \tfrac{1}{u^{(1+\varepsilon)(1-\delta)}}. \end{aligned} \tag{C.10}$$

Similarly to the finite variance case we conclude that if (C.4) holds, then

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\} \mid X \leq u^{1-\delta}\right] \leq \exp\left(\gamma^2 \cdot E|X|^{1+\varepsilon} \cdot \frac{\exp(\gamma)}{(1-\varepsilon)} \cdot \frac{1}{u^{(1+\varepsilon)(1-\delta)}} + 2E|X|^{1+\varepsilon} \cdot \frac{1}{u^{(1+\varepsilon)(1-\delta)}}\right),$$

which in turn yields (C.2). The last part of the result, namely (C.7) follows from elementary algebra and the fact that we are requiring $u \geq 1$. □

D Proof of Lemma 5

Proof. Notice that

$$\begin{aligned} P\left(B_k^c\right) &\leq \sum_{j=n_{k-1}}^{n_k-1} P\left(X_j > (j\mu + m)^{1-\delta}\right) \\ &\leq n_k P\left(X_1 > (n_{k-1}\mu + m)^{1-\delta}\right). \end{aligned}$$

Now we can continue and apply the same arguments as in Lemma 3 to conclude the proof. □

References

[1] S. Asmussen. *Applied Probability and Queues, 2nd ed.* Springer, New York, 2003.

[2] J. Blanchet and X. Chen. Steady-state simulation for reflected Brownian motion and related networks. *http://arxiv.org/pdf/1202.2062.pdf*, 2012.

[3] J. Blanchet and J. Dong. Sampling point processes on stable unbounded regions and exact simulation of queues. In *Proceedings of the 2012 Winter Simulation Conference C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds*, 2012.

[4] J. Blanchet and K. Sigman. On exact sampling of stochastic perpetuities. *Journal of Applied Probability*, 48A:165–182, 2011.

[5] S. Connor and S. Kendall. Perfect simulation of M/G/c queues. *http://arxiv.org/abs/1402.7248*, 2014.

[6] R. Durrett. *Probability: Theory and Examples.* Duxbury Advanced Series, 2005.

[7] K. B. Ensor and P. W. Glynn. Simulating the maximum of a random walk. *Journal of Statistical Planning and Inference*, 85:127–135, 2000.

[8] S. Foss and A. Sapozhnikov. Convergence rates in monotone separable stochastic networks. *Queueing Systems*, 52(2):125–137, 2006.

[9] J. Dong J. Blanchet and Y. Pei. Perfect sampling of GI/G/c queues. *Submitted*, 2015.

[10] W. S. Kendall. Perfect simulation for the area-interaction point process. In *Probability Towards 2000 (ed. L. Accardi and C.C. Heyde). Lecture Notes in Statistics.* volume 128, New York; Springer-Verlag, 218-234, 1998.

[11] J. F. C. Kingman and M. F. Atiyah. The single server queue in heavy traffic. *Proceedings of the Cambridge Philosophical Society*, 57:902, 1961.

[12] K. Murthy, S. Juneja, and J. Blanchet. State-independent importance sampling for random walks with regularly varying increments. *http://arxiv.org/pdf/1206.3390.pdf*, 2013.

[13] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms (Atlanta, Georgia: Proceedings of the Seventh International Conference on Random Structures and Algorithms)*, 9:223–252, 1996.

[14] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue. *Journal of Applied Probability*, 48A:209–216, 2011.

[15] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue: The general case for $\rho < c$. *Queueing Systems*, 70:37–43, 2012.