Stochastic Systems

arXiv: arXiv:0000.0000

ALGORITHMS FOR THE UPPER BOUND MEAN WAITING TIME IN THE $GI/GI/1$ QUEUE

By Yan Chen*, and Ward Whitt*

October 27, 2018

Effective numerical and simulation algorithms are developed to compute the tight upper bound of the mean steady-state waiting time in the $GI/GI/1$ queue given the first two moments of the interarrival-time and service-time distributions. The upper bound is attained asymptotically by two-point distributions as the upper mass point of the service-time distribution increases and the probability decreases, while one mass of the interarrival-time distribution is fixed at 0. The algorithms are aided by reductions of these special queues to $D/GI/1$ and $GI/D/1$ models. One numerical algorithm exploits a negative binomial recursive formula, while another exploits a discrete-time Markov chain recursion. For simulations, in order to address the rare event associated with the large service time, a key step is to exploit the representation of the mean waiting time in terms of the idle-time distribution, which is insensitive to the rare event of the large service time. The computational efficiency of different methods is compared..

1. Introduction. In this paper we study numerical and simulation algorithms for calculating the tight upper bound for the mean steady-state waiting time in the $GI/GI/1$ queue with unlimited waiting room and the first-come first-served service discipline, where the interarrival-time and service-time distributions are partially characterized by their first two moments. This bound can be used as a conservative (worst case) approximation or, combined with the known lower bound, to determine the range of possible values, which is useful in evaluating the quality of approximations.

This paper is a sequel to [5], which presented theoretical and numerical evidence implying that the upper bound is attained asymptotically by two-point interarrival-time and service-time distributions as the upper mass point of the service-time distribution increases and the probability decreases (with the other mass approaching the mean), while one mass of the interarrival-time distribution is fixed at 0. There is a long-standing interest in these upper bounds, starting in [10] and continuing in [7], [8], [21] and the references there. The algorithms here are also of interest because they

*Department of Industrial Engineering and Operations Research, Columbia University

Keywords and phrases: the single-server queues, bounds for the mean waiting time, extremal queues, two-point distributions, stochastic simulation

are based on three different convenient alternative representations for the mean waiting time $\mathbb{E}[W]$ in the $F_0/G_{u^*}/1$ extremal model.

1.1. *The GI/GI/1 Model.* There is a sequence of independent and identically distributed (i.i.d.) interarrival times $\{U_n : n \geq 1\}$ each distributed as U with cumulative distribution function (cdf) F, which is independent of a sequence of i.i.d. service times $\{V_n : n \geq 1\}$, each distributed as V with cdf G. Let an interarrival time U have mean $\mathbb{E}[U] \equiv \lambda^{-1}$ and squared coefficient of variation (scv, variance divided by the square of the mean) c_a^2; let a service time V have mean $\mathbb{E}[V] \equiv \tau$ and scv c_s^2. Assume that the second moments exist, so that the scv's c_a^2 and c_s^2 and the means are finite as well. Assume that $\rho \equiv \lambda\tau < 1$, so that the model is stable. By choosing measuring units, we let $\lambda = 1$, so that $\tau = \rho$.

Let W_n be the waiting time of customer n, i.e., the time from arrival until starting service, assuming that the system starts empty with $W_0 \equiv 0$, where $\equiv$ denotes equality by definition. The sequence $\{W_n : n \geq 0\}$ satisfies the Lindley recursion

$$W_{n+1} = [W_n + V_n - U_n]^+, \quad n \geq 0, \tag{1.1}$$

where $x^+ \equiv \max\{x, 0\}$, V_n is the service time of customer n, U_n is the interarrival time between customers n and $n+1$, and a 0^{th} customer arrives at time 0 to find an empty system.

The waiting time of customer n, starting with an empty system, has mean

$$\mathbb{E}[W_n] = \sum_{k=1}^{n} \frac{\mathbb{E}[S_k^+]}{k} < \infty, \tag{1.2}$$

where $S_k \equiv X_1 + \cdots + X_k$ and $X_k \equiv V_k - U_k$, $k \geq 1$, while the steady-state waiting time W has mean equal to the associated infinite sum, which converges under the finite second moment assumption; e.g., see §§X.1-X.2 of [3] or (13) in §8.5 of [6].

For numerical computation of $\mathbb{E}[W]$, formula (1.2) is unattractive, because it indicates that we need to calculate an infinite sum of terms, each of which involves a k-fold convolution integral. Effective algorithms avoid that computational approach. One way to proceed is to apply numerical transform inversion with the Pollaczek contour integral representation, as in (5) of [1], i.e.,

$$\mathbb{E}[W] = \frac{1}{2\pi i} \int_C \log\{1 - \phi(z)\} \frac{dz}{z}, \tag{1.3}$$

where $i \equiv \sqrt{-1}$, z is a complex variable,

$$\phi(z) \equiv \mathbb{E}[e^{z(V-U)}] \tag{1.4}$$

and C is a contour in the complex plane to the left of, and parallel to, the imaginary axis, and to the right of any singularities of $log\{1-\phi(z)\}$ in the left half plane. As a regularity condition, we assume that the transform ϕ in (1.4) is analytic in the complex plane for z is the strip $|z| < \delta$ for some $\delta > 0$. As in many probability applications, convolution is avoided by considering the transform in (1.4).

Unfortunately, our model with two-point distributions does not satisfy the regularity condition; e.g., see §14 of [2]. As shown in [1], that difficulty can be avoided by asymptotic arguments. That was illustrated by calculating the cumulants and distribution of W in the $E_k/E_k/1$ for a wide range of k, even up to $k = 10^4$. In this paper, we will derive model reductions that will also enable us to avoid direct convolution in other ways.

1.2. *Bounds and Extremal $GI/GI/1$ Queues.* The classical upper bound (UB) for the steady-state mean is the Kingman [10] bound,

$$\mathbb{E}[W] \leq \frac{\rho^2([c_a^2/\rho^2] + c_s^2)}{2(1-\rho)}. \tag{1.5}$$

An improvement is provided by the Daley [7] UB, which replaces the term c_a^2/ρ^2 by $(2-\rho)c_a^2/\rho$, i.e.,

$$\mathbb{E}[W] \leq \frac{\rho^2([(2-\rho)c_a^2/\rho] + c_s^2)}{2(1-\rho)}. \tag{1.6}$$

Both of these bounds are asymptotically correct in heavy traffic, i.e.,

$$\lim_{\rho\to 1}(1-\rho)\mathbb{E}[W](\rho)] = (c_a^2 + c_s^2)/2. \tag{1.7}$$

In fact, the heavy-traffic limit does much more, showing that the scaled waiting-time distribution is asymptotically exponential and thus is asymptotically fully characterized by its mean.

Theorem 4 of [5] shows, for distributions with bounded support, that the UB is attained at interarrival-time and service-time distributions each with support on at most three points. Afterwards, [5] conducts numerical optimization and simulations within that special class of distributions to show that the UB is attained by $\mathbb{E}[W(F_0, G_{u^*})]$, where F_0 is the two-point distribution with one mass on 0, while G_{u^*} is shorthand for the limit of

$\mathbb{E}[W(F_0, G_u)]$ as $M_s \to \infty$, where G_u is the two-point distribution with one mass at the upper boundary M_s. The purpose of this paper is to present algorithms to efficiently calculate or estimate $\mathbb{E}[W(F_0, G_{u^*})]$.

To elaborate, the UB interarrival-time cdf with mean m_1 and second moment $m_2 = m_1^2(c_a^2 + 1)$, referred to here as F_0, is attained at the two-point interarrival-time distribution with probability mass $c_a^2/(1+c_a^2)$ at 0 and probability mass $1/(c_a^2 + 1)$ at $(m_2/m_1) = m_1(c_a^2 + 1)$. The LB interarrival-time cdf, referred to here as F_u is attained at the two-point interarrival-time distribution with probability mass $c_a^2/(c_a^2 + (r-1)^2)$ at M_a, the upper bound of the support, and mass $(r-1)^2/(c_a^2 + (r-1)^2)$ on $1 - c_a^2/(r-1)$ for $r \equiv M_a/m_1$. (For these, we scale so that $m_1 = 1$. We use the notation G_0 and G_u for the corresponding service-time cdf's G with support $[0, M_s]$, where scale so that $m_1 = \rho$.)

Under the assumption that $\mathbb{E}[W(F_0, G_{u^*})]$ is indeed the tight UB, Theorem 1 of [5] provides a new UB, which is an improvement to the UB formulas in (1.5) and (1.6), namely,

$$\mathbb{E}[W(F_0, G_{u^*})] \le \frac{2(1-\rho)\rho/(1-\delta)c_a^2 + \rho^2 c_s^2}{2(1-\rho)}, \tag{1.8}$$

where $\delta \in (0,1)$ and $\delta = \exp(-(1-\delta))/\rho))$. Tables 1 and 2 of [5] show that the new bound (1.8) is very accurate, but it is not tight.

The lower bound (LB), which has long been known, see [16], §5.4 of [15], §V of [18], Theorem 3.1 of [8] and references there, has explicit formula

$$\mathbb{E}[W(LB)] = \frac{\rho^2((1 + c_s^2)\rho - 1)^+}{2(1-\rho)}, \tag{1.9}$$

The LB is not attained at a two-point distribution. The LB is attained asymptotically by $D/A_3/1$ distribution as $M_a \to \infty$, where A_3 denotes any three-point service-time distribution that concentrates all mass on nonnegative-integer multiples of the deterministic interarrival time. Of course, the deterministic distribution does not have the given scv c_a^2 (unless $c_a^2 = 0$); the LB arises as the limit of two-point interarrival-time distributions with one mass approaching the mean from below, while the other mass point grows (and associated probability decreases).

1.3. *Efficient Algorithms.* Our purpose in this paper is to develop and evaluate algorithms to compute $\mathbb{E}[W]$ in the extremal $F_0/G_{u^*}/1$ queue. That involves computing the limit of $\mathbb{E}[W]$ in the $F_0/G_u/1$ model with finite support as $M_s \to \infty$. This is challenging because the large service time is a

rare event. For example, simulating the Lindley recursion via inverse method is not so effective to estimate $\mathbb{E}[W]$ accurately.

In §2 we first show that the tight UB $\mathbb{E}[W(F_0, G_{u^*})]$ provides a significant improvement over previous bounds by comparing the estimates of the tight UB associated with $c_a^2 = c_s^2 = 4.0$ and $c_a^2 = c_s^2 = 0.5$, as estimated by the [12] simulation algorithm. Then we show that effective algorithms can be developed if we transform the problem. In §3 we introduce our first model reduction. Drawing on [9] or [17], we show that the mean waiting time in any $F_0/G/1$ model can be expressed in terms of the mean waiting time in an associated $D/G/1$ model with a new service-time distribution. Then, drawing on [8], in §4 we introduce a second model reduction. We show that the mean waiting time in any $F/G_{u^*}/1$ model can be expressed in terms of the mean waiting time in an associated $F/D/1$ model. In §5 we use the first representation to produce the first effective numerical algorithm involving the negative binomial distribution.

For further progress, following [11], [12] and [21], in §6 we review the representation of the mean waiting time $\mathbb{E}[W]$ in terms of the parameter vector $(1, c_a^2, \rho, c_s^2)$ and the idle-time distribution. When combined with the idle-time representation, this yields other convenient ways calculate or estimate $\mathbb{E}[W]$ via numerical algorithms and simulations. In §7 we develop an algorithm for computing the first two moments of the idle-time distribution based on the first passage time in a finite-state discrete-time Markov chain. We then study three simulation algorithms in §8 and draw conclusions in §9.

2. A Comparison of Different Bounds and Approximations. To show that the new UB $\mathbb{E}[W(F_0, G_{u^*})]$ provides a significant improvement, we compare the estimates of the tight UB for in the $GI/GI/1$ model with given first two moments associated with $c_a^2 = c_s^2 = 4.0$ and $c_a^2 = c_s^2 = 0.5$, as estimated by the [12] simulation algorithm, to other bounds and approximations in Tables 1 and 2. Comparisons for the associated mixed cases $c_a^2 = 4.0, c_s^2 = 0.5$ and $c_a^2 = 0.5, c_s^2 = 4.0$ appear in Tables 16 and 17.

The estimated UB is the "Tight UB" in these tables, while the LB is (1.9), the new UB is (1.8), the [7] bound is (1.6) and the [10] bound is (1.5). The common heavy-traffic (HT) approximation is

$$\mathbb{E}[W] \approx \frac{\rho^2(c_a^2 + c_s^2)}{2(1-\rho)}. \tag{2.1}$$

The MRE is the maximum relative error between the new bound in (1.8) and the estimated tight UB. Example 1 in §2 of [5] shows that this new UB is not tight for $GI/M/1$, using exact calculations as in [18].

TABLE 1
A comparison of the bounds and approximations for the steady-state mean $\mathbb{E}[W]$ *as a function of* ρ *for the case* $c_a^2 = c_s^2 = 4.0$ *and* $c_s^2 = 4.0$.

ρ	Tight LB (1.9)	HTA (2.1)	Tight UB	UB Approx (1.8)	δ	MRE	Daley (1.6)	Kingman (1.5)
0.10	0.00	0.044	0.422	0.422	0.000	0.003%	0.44	2.24
0.20	0.00	0.200	0.904	0.906	0.007	0.19%	1.00	2.60
0.30	0.00	0.514	1.499	1.51	0.041	0.60%	1.71	3.11
0.40	0.00	1.07	2.304	2.33	0.107	0.94%	2.67	3.87
0.50	0.25	2.00	3.470	3.51	0.203	1.15%	4.00	5.00
0.60	1.00	3.60	5.295	5.35	0.324	1.07%	6.00	6.80
0.70	2.42	6.53	8.441	8.52	0.467	0.93%	9.33	9.93
0.80	5.50	12.80	14.92	15.02	0.629	0.67%	16.00	16.40
0.90	15.25	32.40	34.72	34.84	0.807	0.35%	36.00	36.20
0.95	35.13	72.20	74.62	74.76	0.902	0.18%	76.00	76.10
0.98	95.05	192.1	194.6	194.7	0.960	0.07%	196.0	196.0
0.99	195.0	392.0	394.5	394.7	0.980	0.04%	396.0	396.0

TABLE 2
A comparison of the bounds and approximations for the steady-state mean $\mathbb{E}[W]$ *as a function of* ρ *for the case* $c_a^2 = c_s^2 = 0.5$.

ρ	Tight LB (1.9)	HTA (2.1)	Tight UB	UB Approx (1.8)	δ	MRE	Daley (1.6)	Kingman (1.5)
0.10	0.00	0.006	0.053	0.053	0.000	0.04%	0.056	0.281
0.20	0.00	0.025	0.113	0.113	0.007	0.53%	0.125	0.325
0.30	0.00	0.064	0.184	0.189	0.041	2.35%	0.214	0.389
0.40	0.00	0.133	0.280	0.291	0.107	3.82%	0.333	0.483
0.50	0.00	0.250	0.414	0.439	0.203	5.71%	0.500	0.625
0.60	0.00	0.450	0.637	0.669	0.324	4.78%	0.750	0.850
0.70	0.00	0.817	1.017	1.060	0.467	4.53%	1.17	1.24
0.80	0.00	1.600	1.822	1.877	0.629	2.95%	2.00	2.05
0.90	1.08	4.050	4.295	4.355	0.807	1.38%	4.50	4.53
0.95	3.54	9.03	9.284	9.344	0.902	0.65%	9.50	9.51
0.98	11.0	24.0	24.27	24.34	0.960	0.27%	24.5	24.5
0.99	23.5	49.0	49.27	49.34	0.980	0.14%	49.5	49.5

From these tables, we see that the range $UB - LB$ is remarkably wide, which largely can be explained by the LB, which does not depend on the arrival scv c_a^2. We also see that the heavy-traffic approximation and all the UBs tend to agree in HT, but not in light traffic. Moreoever, we see significant improvement going from the [10] bound in (1.5) to the [7] bound in (1.6) to the new UB in (1.8). The MRE in the [7] bound for these cases is about 14% at $\rho = 0.5$.

In closing this section, we emphasize that it remains to prove: (i) that (1.8) is a legitimate UB and (ii) that the mean $\mathbb{E}[W(F_0, G_{u^*})]$ estimated for the tight UB here is indeed the tight UB. Theorem 1 of [5] proves (i) under the assumption that (ii) is correct. Nevertheless, we have provided strong numerical evidence that the $F_0/G_{u^*}/1$ model yields the tight UB. If that can be accepted, then formula (1.8) serves as an excellent approximation formula.

3. The Reduction of $F_0/GI/1$ to $D/GI/1$. In this section we show that, for any service-time cdf G, the mean waiting time in the $F_0/GI/1$ queue can be expressed in terms of the mean waiting time in an associated $D/G/1$ queue with a new service-time distribution. The key observation is that the $F_0/G/1$ queue corresponds to the $D/G/1$ queue with batch arrivals; then the new service-time cdf is the sum of the service times in the batch. However, we need to do other adjustments as well.

Let F_0 be the two-point upper bound extremal distribution with mean 1 and mass $p \equiv 1/(c_a^2+1)$ on c_a^2+1 and mass $1-p$ on 0. Let $RS(V,p)$ be a random variable distributed as

$$RS(V,p) \stackrel{\mathrm{d}}{=} \sum_{k=1}^{N(p)} V_k, \tag{3.1}$$

where $N(p)$ is a geometric random variable on the positive integers, having mean $\mathbb{E}[N(p)] = 1/p$ and $\{V_k : k \geq 1\}$ is a sequence of i.i.d. random variables distributed as a service time V. Let $D(x)$ be a deterministic random variable assuming the constant value x. For the interarrival times, we will consider $x = 1/p = (c_a^2+1)$.

THEOREM 3.1. *For the $F_0(p)/GI/1$ model with service time V having mean ρ and scv c_s^2, the mean steady-state waiting time can be expressed as*

$$\begin{aligned} \mathbb{E}[W(F_0(p)/GI/1)] &= \mathbb{E}[W(D(1/p)/RS(V,p)/1)] + (\mathbb{E}[N(p)]-1)\mathbb{E}[V] \\ &= \mathbb{E}[W(D(1/p)/RS(V,p)/1)] + \rho(1-p)/p \\ &= \mathbb{E}[W(D(1/p)/RS(V,p)/1)] + \rho c_a^2. \end{aligned} \tag{3.2}$$

where $RS(V,p)$ is the geometric random sum in (3.1).

Proof.. The F_0 interarrival time means that a random number of arrivals, distributed as $N(p)$, arrive at deterministic intervals with deterministic value $1/p = c_a^2+1$. So the model has batch arrivals. The result in (3.2) follows from [9] or Theorem 1 of [17], which states that the delay of an arbitrary customer in the batch is distributed the same as the delay of the last customer in the batch when the batch-size distribution is geometric. Because $\mathbb{E}[W(D(1/p)/RS(V,p)/1)]$ is the expected delay of the first customer in a batch, we need to add the second term in (3.2) to get the delay of the last customer in the batch; e.g., see §III of [17]. ∎

To work with the $D(1/p)/RS(V,p)/1$ model, we want the mean and variance of the random sum $RS(V,p)$ in (3.1).

LEMMA 3.1. (*random sum moments*) *Given that V has mean ρ and scv c_s^2, the mean and variance of the random sum $RS(V,p)$ in (3.1) are*

$$\mathbb{E}[RS(V,p)] = \mathbb{E}[N(p)]\mathbb{E}[V] = \frac{\rho}{p} = \rho(c_a^2+1) \tag{3.3}$$

and

$$Var(RS(V,p)) = \rho^2 c_s^2(c_a^2+1) + \rho^2 c_a^2(1+c_a^2). \tag{3.4}$$

Hence,

$$\bar{c}_s^2 \equiv \frac{Var(RS(V,p))}{\mathbb{E}[RS(V,p)]^2} = \frac{\rho^2 c_s^2(c_a^2+1) + \rho^2 c_a^2(1+c_a^2)}{\rho^2(1+c_a^2)^2} = \frac{c_a^2+c_s^2}{1+c_a^2}. \tag{3.5}$$

Proof.. We apply the standard formulas for random sums from p. 113 of [14]. For the variance,

$$\begin{aligned} Var(RS(V,p)) &= Var(V)\mathbb{E}[N] + (\mathbb{E}[V])^2 Var(N) = \frac{\rho^2 c_s^2}{p} + \frac{\rho^2(1-p)}{p^2} \\ &= \rho^2 c_s^2(1+c_a^2) + \rho^2 c_a^2(1+c_a^2), \end{aligned} \tag{3.6}$$

as claimed. ∎

THEOREM 3.2. *For the $D(1/p)/RS(V,p)/1$ model, the Kingman upper bound on the mean steady-state waiting time is*

$$\begin{aligned} \mathbb{E}[W(D(p)/RS(V,p)/1)] &\le \frac{\rho\mathbb{E}[RS(V,p)]((\bar{c}_a^2/\rho^2)+\bar{c}_s^2)}{2(1-\rho)} \\ &= \frac{\rho^2(1+c_a^2)\bar{c}_s^2}{2(1-\rho)} = \frac{\rho^2(c_a^2+c_s^2)}{2(1-\rho)}. \end{aligned} \tag{3.7}$$

Hence, the associated upper bound for the $F_0(p)/GI/1$ model is

$$\mathbb{E}[W(F_0/GI/1)] \le \frac{\rho^2(c_a^2+c_s^2)}{2(1-\rho)} + \rho c_a^2 = \frac{\rho^2(Ac_a^2+c_s^2)}{2(1-\rho)}, \tag{3.8}$$

where

$$A \equiv A(\rho, c_a^2) \equiv 1 + \frac{2(1-\rho)}{\rho} = \frac{2}{\rho} - 1, \tag{3.9}$$

which makes (3.8) coincide with the Daley [7] bound.

Proof. We exploit Theorem 3.1, which provides the representation (3.2). Then observe, with the aid of Lemma 3.1, that the [10] bound for $\mathbb{E}[W(D(1/p)/RS(V,p)/1)]$ is given by the first term on the first line of (3.8). ∎

4. The Reduction of $GI/G_{u^*}/1$ to $GI/D/1$. Daley proposed another decomposition that can be used to avoid the rare event of the large service time M_s. It allows us to reduce the model $F/G_{u^*}/1$ to $F/D/1$ for arbitrary F. It is reviewed in (10.2) of [8] without proof, referring to an unpublished manuscript. Let D_m denote a deterministic cdf with mass 1 on m.

THEOREM 4.1. (*the Daley decomposition in* (10.2) *of* [8]) *Consider the* $GI/G_u/1$ *model with arbitrary interarrival-time cdf* F *and two-point service-time cdf* $G_u \equiv G_u(M_s)$. *Then*

$$\begin{aligned} \lim_{M_s\to\infty} \mathbb{E}[W(F,G_u)] &= \mathbb{E}[W(F,D_\rho)] + \lim_{M_s\to\infty} \mathbb{E}[W(D_1,G_u)]. \\ &= \mathbb{E}[W(F,D_\rho)] + \frac{\rho^2 c_s^2}{2(1-\rho)}. \end{aligned} \tag{4.1}$$

Proof.. We only give a brief overview. We do a regenerative analysis to compute the mean waiting time, looking at successive busy cycles starting empty. We exploit the classic result that the steady-state mean waiting time is the expected sum of the waiting times over one cycle divided by the expected length of one cycle; e.g., see §3.6 and §3.7 of [13].

As M_s increases, the two-point cdf $G_u \equiv G_u(M_s)$ necessarily places probability of order $O(1/M_s^2)$ on M_s and the rest of the mass on a point just less than the mean service time, ρ. For very large M_s, there will be only rarely, with probability of order $O(1/M_s^2)$, a large service time of order $O(M_s)$. In the limit, most customers never encounter this large service time, so that we get a contribution to the overall mean $\mathbb{E}[W]$ corresponding to $\mathbb{E}[W(F,D_\rho)]$ in the first term on the right in (4.1).

On the other hand, the total impact of the very large waiting time of order M_s is roughly the area of the triangle with height $O(M_s)$ and width $O(M_s)$, which itself is $O(M_s^2)$. When combined with the $O(1/M_s^2)$ probability, this produces an additional $O(1)$ impact on the steady-state mean, which is given by the second term on the right in (4.1). Moreover, because we can use a law-of-large-numbers argument to treat this large service time, the asymptotic impact of that large service time is independent of the interarrival-time cdf beyond its mean, so we can substitute D_1 for the original interarrival-time cdf F with mean 1 in the second term. ∎

COROLLARY 4.1. (*decomposition of the upper bound*) *For the* $GI/GI/1$ *model with extremal interarrival-time cdf* F_0 *and extremal service-time cdf* G_{u^*},

$$\mathbb{E}[W(F_0, G_{u^*})] \equiv \lim_{M_s \to \infty} \mathbb{E}[W(F_0, G_u)] \quad = \quad \mathbb{E}[W(F_0, D_\rho)] + \frac{\rho^2 c_s^2}{2(1-\rho)}.$$

Corollary 4.1 implies that calculating the UB of $\mathbb{E}[W]$ is equivalent to calculating $F_0/D/1$, which has deterministic service time. Clearly, this makes the UB much easier to estimate by classical simulation methods.

COROLLARY 4.2. (*tightness of Kingman's bound*) *For the* $GI/GI/1$ *model with interarrival-time cdf* D *and extremal service-time cdf* G_{u^*},

$$\mathbb{E}[W(F_0, G_{u^*})] \equiv \lim_{M_s \to \infty} \mathbb{E}[W(D, G_u)] = \mathbb{E}[W(D_1, D_\rho)] + \frac{\rho^2 c_s^2}{2(1-\rho)} = \frac{\rho^2 c_s^2}{2(1-\rho)},$$

so that Kingman's bound is asymptotically attained by $D/G_u(M_s)/1$ *as* $M_s \to \infty$.

Finally, we can combine Theorem 3.1 and Corollary 4.1 to obtain

COROLLARY 4.3. (*overall decomposition of the upper bound*) *For the* $GI/GI/1$ *model with extremal interarrival-time cdf* F_0 *and extremal service-time cdf* G_{u^*},

$$\mathbb{E}[W(F_0, G_{u^*})] \quad = \quad \mathbb{E}[W(D(1/p)/RS(D(\rho), p)/1)] + \rho c_a^2 + \frac{\rho^2 c_s^2}{2(1-\rho)}.$$

5. The Negative Binomial Numerical Algorithm. In this section we apply Corollary 4.3 to obtain an efficient algorithm for computing the UB $\mathbb{E}[W(F_0, G_{u^*})]$. Corollary 4.3 implies that it suffices to compute $\mathbb{E}[W]$ in the $D(1/p)/RS(D(\rho), p)/1$ model. The representation of the service time as a geometric random sum allows us to express $\mathbb{E}[W]$ directly in terms of the negative binomial (NB) distribution, without having to perform any convolutions.

Let $NB(n, p)$ be a conventional negative binomial random variable with parameter pair (n, p) for nonnegative integer n and $0 < p < 1$, which has probability mass function (pmf)

$$p_k(n, p) \equiv P(NB(n, p) = k) \equiv \left(\frac{(n+k-1)!}{k!(n-1)!} \right) (1-p)^n p^k, \quad n \geq 0, \tag{5.1}$$

with mean and variance

$$\mathbb{E}[NB(n,p)] = \frac{np}{1-p} \quad \text{and} \quad Var(NB(n,p)) = \frac{np}{(1-p)^2}. \tag{5.2}$$

As often with the NB pmf, because of the factorials, it is convenient to use a recursive algorithm for computation. In the first version we initialize the recursion at $k = 0$, letting $\mathbb{P}(NB(n, 1-p) = 0) = p^n$. Then, we can apply the recursion

$$\mathbb{P}(NB(n,1-p) = k) = \mathbb{P}(NB(n,1-p) = k-1)(n+k-1)/k)(1-p), \tag{5.3}$$

where $p = 1/(1 + c_a^2)$.

However, for the parameter $p = 1/(c_a^2 + 1)$ already defined by F_0, we end up with negative binomial parameter $1 - p$. Let $\lfloor x \rfloor$ be the greatest integer less than or equal to x.

LEMMA 5.1. *(NB representation) For the $D(1/p)/RS(D(\rho), p)/1$ model,*

$$S_n \stackrel{\mathrm{d}}{=} \rho(NB(n, 1-p) + n) - (n/p), \tag{5.4}$$

for S_n in (1.2), so that

$$\begin{aligned} \mathbb{E}[W] &= \rho \sum_{n=1}^{\infty} n^{-1} \mathbb{E}[(NB(n,1-p) + n - (n/p\rho))^+] \\ &= \rho \sum_{n=1}^{\infty} n^{-1} \sum_{k=0}^{\infty} P(NB(n,1-p) = k)(cn - n + k)^+ \\ &= \rho \sum_{n=1}^{\infty} n^{-1} \sum_{k=0}^{\infty} P(NB(n,1-p) > \lfloor cn \rfloor - n + k) \end{aligned} \tag{5.5}$$

for $c \equiv 1/p\rho > 1$, from which $\mathbb{E}[W(F_0/G_{u^*}/1)]$ is obtained from Corollary 4.3.

Proof. First, note that our geometric random variable $N(p)$ in (3.1) takes values in the positive integers, while NB takes values in the nonnegative integers. Recall that the sum of n i.i.d. geometric random variables is negative binomial, so that the connection to our geometric random variable $N(p)$ on the positive integers is $N(p) - 1 \stackrel{\mathrm{d}}{=} NB(1, 1-p)$. Then, for i.i.d. variables $N_k(p) \stackrel{\mathrm{d}}{=} N(p)$,

$$N_1(p) + \cdots + N_n(p) \stackrel{\mathrm{d}}{=} n + NB(n, 1-p). \tag{5.6}$$

Hence, the partial sums S_n in §1.1 satisfies (5.4) so that we obtain (5.5) by (1.2). For the second line in (5.5), we use the representation of the mean in terms of the complementary cdf, as on p. 46 of [13]. ∎

Appropriately truncated versions of the final double sum in (5.5) can then readily be computed. That is illustrated for the middle display in (5.5) in Algorithm 1 below.

Algorithm 1 Basic Negative Binomial Recursion (k in outer loop)

Initially set $\mathbb{E}[W] \leftarrow \rho c_a^2 + \frac{\rho^2 c_s^2}{2(1-\rho)}$ and $p = (1 + c_a^2)^{-1}$.
for $k \in [K]$ **do**
  $S(k) \leftarrow 0, nbpdf \leftarrow p(1-p)^k$
  for $n \in [n]$ **do**
    $S(k) \leftarrow S(k) + nbpdf \max((n+k)\rho - n/p, 0)/n$
    $nbpdf \leftarrow nbpdf(\frac{n+k}{n})p$
  $\mathbb{E}[W] \leftarrow \mathbb{E}[W] + S(k)$
Output $\mathbb{E}[W]$

To explain Algorithm 1, recall that we are applying Corollary 4.3 to obtain an efficient algorithm for computing the UB $\mathbb{E}[W(F_0, G_{u^*})]$. Thus we initialize by the constant term that depends only on the vector (c_a^2, ρ, c_s^2). We add that to $\mathbb{E}[W(D(1/p)/RS(D(\rho), p)/1)]$, which is computed by the recursion.

It now remains to consider how to do the truncations. First, consider the truncation of the sum on k for given n. For given n,

$$\begin{aligned} \mathbb{E}[NB(n, 1-p)] &\equiv m(n) = \frac{n(1-p)}{p} \quad \text{and} \\ Var(NB(n, 1-p)) &\equiv \sigma^2(n) = \frac{n(1-p)}{p^2}. \end{aligned} \tag{5.7}$$

For large n, $NB(n, 1-p)$ is asymptotically Gaussian by the central limit theorem, so for very large n, only $O(\sqrt{n})$ values of k need be considered. In particular, is should suffice to consider $m(n) - a\sigma(n) \le cn + k \le m(n) + a\sigma(n)$ for, e.g., $a = 8$. However, we need to add a term for small k. For $cn + k \le m(n) - a\sigma(n)$, we let $P(NB(n, 1-p) > \lfloor cn \rfloor + k) = 1$. That means we add $(m(n) - a\sigma(n)) \wedge cn \vee 0$, where $a \wedge b \equiv \min\{a, b\}$ and $a \vee b \equiv \max\{a, b\}$.

Finally, the relevant values of n depend on the traffic intensity ρ and other model parameters. For heavy traffic (large ρ), we can use the approximation (2.1) to estimate the relevant n. Moreover, given that the heavy-traffic limit of the waiting-time distribution is exponential, we can see the relevant range of n.

5.1. *Performance of the Negative Binomial Algorithm.* We set different truncation levels K and N to study the computational accuracy and effort of the Negative Binomial (NB) algorithm.

In the experiment, set the truncation level N =1E+03 and K from 1E+03 to 8E+03 to execute Algorithm 1. (It is good to have k in the outer loop because $p = 1/(1 + c_a^2) < 0.5$.) The results are shown in Table 3 for a range of traffic intensities from $\rho = 0.10$ to $\rho = 0.99$. Also shown for comparison in the last two columns are the simulation estimates from the highly accurate [12] simulation method, as given in Table 10.

For $\rho \leq 0.90$, the recursive algorithm with truncation level $N = 1000, K = 3000$ performs well, but for $\rho \geq 0.95$, the numerical values of $\mathbb{E}[W]$ converge as K increases but are not not close to the simulation results.

TABLE 3
Performance of the Basic Negative Binomial Algorithm with Different Truncation Levels

	Algorithm Procedure 1 with $N = 1000$				Minh and Sorli Algorithm	
$\rho \backslash K$	1E+03	2E+03	4E+03	8E+03	$T = 1E + 07$	95%CI
0.1	0.422229	0.422229	0.422229	0.422229	0.422	7.79E-05
0.2	0.903885	0.903885	0.903885	0.903885	0.904	1.30E-04
0.3	1.499234	1.499234	1.499234	1.499234	1.499	1.71E-04
0.4	2.304105	2.304105	2.304105	2.304105	2.304	1.90E-04
0.5	3.470132	3.470132	3.470132	3.470132	3.470	2.25E-04
0.6	5.294825	5.294825	5.294825	5.294825	5.294	2.43E-04
0.7	8.441305	8.441305	8.441305	8.441305	8.442	3.05E-04
0.8	14.916481	14.916937	14.916937	14.916937	14.917	3.22E-04
0.9	34.276662	34.673925	34.718140	34.718140	34.722	5.17E-04
0.95	66.874413	71.232241	73.264743	73.264743	74.621	7.11E-04
0.98	139.659440	152.638886	162.915010	162.915010	194.556	9.29E-04
0.99	245.012809	262.661919	278.499123	278.499123	394.532	1.45E-03

5.2. *Refinement to the Negative Binomial Algorithm for Heavy-traffic.* The difficulty in heavy traffic occurs because as ρ increases, we need larger values of n. For extremely large n, as is needed in heavy traffic, p^n and $(1 - p)^n$ are eventually very small numbers. That causes the probability to become too small to be represented in the implemented floating point number system. Hence, in heavy traffic the basic recursive algorithm broke down because the large values of n caused underflow.

As when computing the steady-state of the birth-and-death processes, e.g. as in §7 of [20], for very large n we can uncounter underflow problems if we start the recursion at 0, but it can be avoided by starting the recursion elsewhere. We avoid the underflow problem by doing two recursions, one up and the other down, starting from the mean. From the central limit theorem, we know that the NB distribution is approximately Gaussian with a mean near its mode. In particular,

$$NB(n, 1 - p) \approx \mathcal{N}(m(n), \sigma^2(n)) \quad \text{as} \quad n \to \infty. \tag{5.8}$$

for $m(n)$ and $\sigma^2(n)$ in (5.7). Hence for large n suffices to consider only a modest range of k, i.e., of order $O(\sqrt{n})$. As a consequence, for large N, we consider $k \le m(n) + 20\sqrt{N}$ in the implementation.

Here is how we proceed: For fixed $n \le N$, we start from mean in (5.3) and let the $\mathbb{P}(NB(n, 1-p) = n(1-p)/p) = 1$ and then do recursive formula (5.3) up and down separately. Define mean $n(1-p)/p$ by $m(n)$. The two-part recursion going up and down becomes

$$\begin{aligned}
&\mathbb{P}(NB(n,1-p) = m(n)+j) \\
&= \mathbb{P}(NB(n,1-p) = m(n)+j-1)(n+m(n)+j-1)/(m(n)+j)(1-p), \\
&\mathbb{P}(NB(n,1-p) = m(n)-j) \\
&= \mathbb{P}(NB(n,1-p) = m(n)-j+1)/(n+m(n)-j)(m(n)-j+1)/(1-p)
\end{aligned}$$

for $j \ge 1$. Afterwards, we normalize the values that obtained from the above recursion to get probabilities of $P(NB(n, 1-p) = k)$ for any k given n.

As in Algorithm 1, in Algorithm 2 we apply Corollary 4.3 to obtain an efficient algorithm for computing the UB $\mathbb{E}[W(F_0, G_{u^*})]$. Thus we initialize by the constant term that depends only on the vector (c_a^2, ρ, c_s^2).

Algorithm 2 Negative Binomial Recursion (Up and Down from the Mean)

Initially set $\mathbb{E}[W] \leftarrow \rho c_a^2 + \frac{\rho^2 c_s^2}{2(1-\rho)}$, $p = (1+c_a^2)^{-1}$, and $m(n) = n(1-p)/p$.
for $n \in [1, N]$ **do**
  $nbpdf(1, m(n)) \leftarrow 1$
  for $k \in [m(n) - 20\sqrt{N}, m(n)]$ **do**
    $nbpdf(1, k-1) \leftarrow nbpdf(1,k)/(n+k-1)(k)/(1-p)$
  for $k \in [m(n), m(n) + 20\sqrt{N} - 1]$ **do**
    $nbpdf(1, k+1) \leftarrow nbpdf(1,k)(n+k)/(k+1)(1-p)$
  Normalize *nbpdf* to obtain $\mathbb{P}(NB(n, 1-p) = k)$
  $S(n) \leftarrow \sum_k \mathbb{P}(NB(n,1-p) = k) \max((n+k)\rho - n/p, 0)$
  $\mathbb{E}[W] \leftarrow \mathbb{E}[W] + S(n)/n$
Output $\mathbb{E}[W]$

We now carefully compare the negative binomial pmf values generated from the basic recursion (5.3) used in Algorithm 1 with the values obtained in the new up-down recursion used in Algorithm 2 in Table 4. We focus on the terms after $m(n)$ and report the values from the term $m(n)$ to $m(n)+10$.

TABLE 4
Comparison of the basic and up-down recursions for generating values of the negative binomial pmf in Algorithms 1 and 2

k	$n_1 = 10$	$n_2 = 10$	k	$n_1 = 100$	$n_2 = 100$	k	$n_1 = 1000$	$n_2 = 1000$
40	0.0279638	0.0279638	400	0.0089128	0.0089128	4000	0	0.0028207
41	0.0272818	0.0272818	401	0.0088906	0.0088906	4001	0	0.0028200
42	0.0265023	0.0265023	402	0.0088641	0.0088641	4002	0	0.0028192
43	0.0256394	0.0256394	403	0.0088333	0.0088333	4003	0	0.0028182
44	0.0247071	0.0247071	404	0.0087983	0.0087983	4004	0	0.0028170
45	0.0237188	0.0237188	405	0.0087592	0.0087592	4005	0	0.0028158
46	0.0226875	0.0226875	406	0.0087160	0.0087160	4006	0	0.0028144
47	0.0216256	0.0216256	407	0.0086689	0.0086689	4007	0	0.0028128
48	0.0205443	0.0205443	408	0.0086179	0.0086179	4008	0	0.0028111
49	0.0194542	0.0194542	409	0.0085631	0.0085631	4009	0	0.0028093
50	0.0183647	0.0183647	410	0.0085047	0.0085047	4010	0	0.0028074

For $n \leq 100$, the results from the two methods agree to all digits shown, but a significant difference occurs when $n = 1000$. At $n = 1000$, underflow occurs in Algorithm 1, which causes the errors we saw for large ρ in Table 3.

5.3. *Performance Studies for the Refined Negative Binomial Algorithm.* First, Algorithm 2 is also very efficient for $\rho \leq 0.95$. Table 5 shows that the new algorithm is effective if we increase N from $1,000$ to $10,000$ as ρ increases.

TABLE 5
Performance of Algorithm 2 with Different Truncation Levels

	Algorithm 2					Minh and Sorli Algorithm	
$\rho \backslash N$	2E+03	4E+03	8E+03	1.6E+04	2E+04	$T = 1E + 07$	95%CI
0.1	0.422229	0.422229	0.422229	0.422229	0.422229	0.422	7.79E-05
0.2	0.903885	0.903885	0.903885	0.903885	0.903885	0.904	1.30E-04
0.3	1.499234	1.499234	1.499234	1.499234	1.499234	1.499	1.71E-04
0.4	2.304105	2.304105	2.304105	2.304105	2.304105	2.304	1.90E-04
0.5	3.470132	3.470132	3.470132	3.470132	3.470132	3.470	2.25E-04
0.6	5.294825	5.294825	5.294825	5.294825	5.294825	5.294	2.43E-04
0.7	8.441305	8.441305	8.441305	8.441305	8.441305	8.442	3.05E-04
0.8	14.916937	14.916937	14.916937	14.916937	14.916937	14.917	3.22E-04
0.9	34.721476	34.721484	34.721484	34.721484	34.721484	34.722	5.17E-04
0.95	74.552341	74.619631	74.620917	74.620937	74.620937	74.621	7.11E-04

The numerical algorithm is more efficient than the simulation. It requires no more than 30 seconds cpu time in the worse case ($N = 2E+04$, $\rho = 0.95$) ro produce more than 10 decimal places accuracy, while the MS simulation algorithm only attain 1E-04 confidence interval level for $0.5 \leq \rho \leq 0.95$ while producing 3 decimal places accuracy within around 30 seconds cpu times.

Next, we apply Algorithm 2 for the heavy-traffic cases with $\rho = 0.98$ and $\rho = 0.99$. To do so, we restrict the range of k to $k \leq m(n) + 20\sqrt{N}$ for the purpose of setting smaller N. Table 6 below shows that the poor performance of the NB algorithm in Table 3 has been improved dramatically by the alternative algorithm.

TABLE 6
Performance of Algorithm 2 in Heavy Traffic

	Agorithm 2 for Heavy-Traffic				Minh and Sorli Algorithm	
$\rho \backslash N$	1E+04	2E+04	3E+04	4E+04	$T = 1E+07$	
0.98	194.0544167173	194.5385548017	194.5559125683	194.5567071265	194.556	9.29E-04
	5E+04	1E+05	2E+05	3E+05		
0.98	194.5567179973	194.5567742874	194.5567742874	194.5567742874	194.556	9.29E-04
$\rho \backslash N$	1E+04	3E+04	5E+04	1E+05	$T = 1E+07$	
0.99	372.0880005430	372.0880005430	391.8858614678	394.5238008176	394.532	1.45E-03
	2E+05	3E+05	4E+05	5E+05		
0.99	394.5331823499	394.5331886695	394.5331886695	394.5331886695	394.532	1.45E-03

REMARK 5.1. Our experiments suggest that it suffices to set $N = \Theta(1/(1-\rho)^3)$ to obtain highly accurate results.

REMARK 5.2. Since the service-time variabiity parameter c_s^2 is not used in 2, Table 5 and Table 6 can be reused to compute $\mathbb{E}[W(F_0/G_{u^*}/1)]$ with any other c_s^2 via Corollary 4.3.

6. Exploiting the Idle-Time Representation. To develop alternative algorithms, following [11], [12] and [21], we relate the mean waiting time given the first two moments of the interarrival time and service time to the first two moments of the idle time I. In §6.1 we review the basic relation. in §6.2 we discuss the implications of the relation when we let $M_s \to \infty$. In §6.3 we show the advantage of combining Theorem 6.1 and Corollary 4.1. Later, in §7 we apply the representation to develop a new numerical algorithm based on computing absorption probabilities in finite-state discrete-time Markov chains (DTMCs).

6.1. *The Basic Representation.* The key relation is in

THEOREM 6.1. (*the idle-time representation, Theorem 1 of [11]*) *In the $GI/GI/1$ queue with cdf's F and G having parameter 4-tuple $(1, c_a^2, \rho, c_s^2)$,*

$$\mathbb{E}[W] \equiv \mathbb{E}[W(F,G)] = \psi(1, c_a^2, \rho, c_s^2) - \phi(I), \tag{6.1}$$

where

$$\psi(1, c_a^2, \rho, c_s^2) \equiv \frac{\mathbb{E}[(U-V)^2]}{2\mathbb{E}[U-V]} = \frac{\rho^2([c_a^2/\rho^2] + c_s^2)}{2(1-\rho)} + \frac{1-\rho}{2} \tag{6.2}$$

and

$$\phi(I) \equiv \phi(F,G) = \frac{\mathbb{E}[I^2]}{2\mathbb{E}[I]} = \mathbb{E}[I_e], \tag{6.3}$$

with I being the steady-state idle time and I_e being a random variable with the associated stationary excess distribution (as in renewal theory).

Notice that $\mathbb{E}[W]$ depends on the model distributions F and G beyond the parameter vector $(1, c_a^2, \rho, c_s^2)$ only through $\phi(I) = \mathbb{E}[I_e]$ in (6.3). For the $M/GI/1$ model, I is distributed as F, $\phi(I) = 1$ and simple algebra yields the exact Pollaczek-Khintchine formula. In general, the first term on the right in (6.2) is the [10] upper bound. For the [10] bound to be obtained, the second term on the right in (6.2) would have to be exactly cancelled by the second term on the right in (6.1).

6.2. *The Limit as $M_s \to \infty$.* This section is based on the notion that the upper bound is obtained as the limit of $\mathbb{E}[W]$ within the $F_0/G_u/1$ model as $M_s \to \infty$. Because the mean waiting time is not continuous as $M_s \to \infty$, but the idle-time distribution is, we approach the upper bound via the idle time.

We can apply Theorem 3.1 to obtain a limit within the decomposition. For that purpose, let $\phi(I; A, B)$ denote $\phi(I)$ for the model with interarrival time A and service time B. We will consider $A = D(1/p)$ and $B = RS(D(\rho), p)$.

THEOREM 6.2. (*limit within the decomposition*) *For the $F_0/G_u/1$ model with parameter vector $(1, c_a^2, \rho, c_s^2)$ and service-distribution support $[0, M_s]$,*

$$\lim_{M_s \to \infty} \mathbb{E}[W(F_0/G_u/1)] \quad = \quad \psi(1, c_a^2, \rho, c_s^2) - \phi(I; 1, c_a^2, \rho, 0). \tag{6.4}$$

In other words, the first term in (6.1) is independent of M_s and thus is unchanged by the limit on M_s, whereas the second term changes, consistent with the distribution G_u approaching $D(\rho)$, and having the limiting mean but 0 variance. As a consequence,

$$\begin{aligned} \lim_{M_s \to \infty} \mathbb{E}[W(F_0/G_u/1)] &= \psi(1, c_a^2, \rho, c_s^2) + \rho c_a^2 - \lim_{M_s \to \infty} \phi(I) \\ &= \psi(1, c_a^2, \rho, c_s^2) + \rho c_a^2 - \phi(I; D(1/p), RS(D, p)), \\ &= \psi(1, c_a^2, \rho, c_s^2) + \rho c_a^2 - \phi(I; (1 + c_a^2), 0, \rho(1 + c_a^2), \bar{c}_s^2) \end{aligned} \tag{6.5}$$

where $\phi(I; D(1/p), RS(D, p))$ means (6.3) for the $D(1/p), RS(D, p))/1$ model and the parameter vector for that model is $((1 + c_a^2), 0, \rho(1 + c_a^2), \bar{c}_s^2)$ for

$$\bar{c}_s^2 \equiv \frac{c_a^2}{1 + c_a^2}. \tag{6.6}$$

Theorem 6.2 implies that it only remains to evaluate the idle-time term $\phi(I)$ in the last line of (6.5) for the $D(p)/RS(D, p)/1$ model, for which the

only randomness is in the random sum in the service times. The random sum is a geometric random sum of constants in this case. When we apply the [12] method for simulation, it suffices to reduce variance by ignoring the large M_s. We treat the service times as D with mean ρ. But, when we do so, we have to make adjustments in the final formulas as indicated above.

COROLLARY 6.1. (*one waiting time in terms of the other*) *For the $F_0/G_u/1$ model with support bound M_s and parameter vector $(1, c_a^2, \rho, c_s^2)$,*

$$\begin{aligned}\lim_{M_s\to\infty} \mathbb{E}[W(F_0/G_u/1)] &= \mathbb{E}[W(D(1/p)/RS(D(\rho),p)/1)] + \rho c_a^2 + \frac{\rho^2 c_s^2}{2(1-\rho)} \\ &= \mathbb{E}[W(D(1/p)/RS(D(\rho),p)/1)] + \frac{\rho^2(Bc_a^2 + c_s^2)}{2(1-\rho)} \qquad (6.7)\end{aligned}$$

for

$$B \equiv (2/\rho) - 2. \qquad (6.8)$$

6.3. *Combining Theorem 6.1 and Corollary 4.1.* Combining Theorem 6.1 and Corollary 4.1, we obtain

COROLLARY 6.2. (*reduction to idle time*) *For the $GI/GI/1$ model with extremal interarrival-time cdf F_0 and extremal service-time cdf G_{u^*},*

$$\begin{aligned}\mathbb{E}[W(F_0, G_{u^*})] &\equiv \lim_{M_s\to\infty} \mathbb{E}[W(F_0/G_u/1)] \\ &= \frac{c_a^2 + \rho^2 c_s^2}{2(1-\rho)} + \frac{1-\rho}{2} - \phi(I; 1, c_a^2, \rho, c_s^2), \qquad (6.9)\end{aligned}$$

where I is the idle time in an $F_0/G_{u^*}/1$ queue or, equivalently, in a $F_0/D/1$ queue for an appropriate D.

Corollary 6.2 shows that to determine the UB $\mathbb{E}[W(F_0/G_{u^*}/1)]$, it suffices to calculate the term $\phi(I; 1, c_a^2, \rho, c_s^2)$ in (6.3) for the $F_0/D/1$ model via effective algorithms. In contrast, Theorem 6.2 concludes that it suffices to calculate ϕ in (6.3) for the $D/RS(D(\rho), p)/1$ model, but we see that these are equivalent, because we can go from one to the other by applying Theorem 3.1. Thus we conclude that §3 and §4 are two different ways to reach essentially the same conclusion.

7. Computing the Distribution and Moments of the Idle Time. Theorem 6.2 implies that the steady-state mean waiting time $\mathbb{E}[W]$ in the extremal $F_0/G_{u^*}/1$ model can be expressed in terms of the first two moments of the steady-state idle time I in the $D(1/p)/RS(D,p)/1$ model and the parameter vector $(1, c_a^2, \rho, c_s^2)$. In this section we show how to develop algorithms to calculate the distribution and moments of I in the $D(1/p)/RS(D,p)/1$ model based on a random walk representation.

7.1. *A Random Walk Absorption Representation of the Idle-Time.* For the reduced model $D(1/p)/RS(D,p)/1$, the steady-state idle time can be expressed in terms of a random walk $\{Y_k : k \geq 0\}$ defined in terms of the recursion,

$$Y_{k+1} = Y_k + \rho N_k - (1 + c_a^2), \quad k \geq 1, \quad Y_0 \equiv 0. \tag{7.1}$$

The random variables $\rho N_k - (1+c_a^2)$ are the steps of the random walk. Each step is the net input of work from one arrival time to the next. Because N_k take values on the positive integers, the possible steps are $k\rho - (1 + c_a^2)$ for $k \geq 1$, so that $\rho N_k - (1 + c_a^2) \geq \rho - (1 + c_a^2)$.

As long as $Y_k \geq 0$, Y_k represents the work in the system at the time of the k^{th} arrival, starting empty. The number of customers served in that busy cycle, N_c, and the length of a busy cycle, C, are then

$$N_c = \inf\{k \geq 1 : Y_k \leq 0\} \quad \text{and} \quad C = N_c(1 + c_a^2). \tag{7.2}$$

The associated idle-time random variable is distributed as

$$I \stackrel{\text{d}}{=} -Y_{N_c}, \quad \text{so that} \quad 0 \leq I \leq c_a^2 + 1 - \rho. \tag{7.3}$$

7.2. *An Idle-Time Simulation Algorithm.* Given N i.i.d. copies of I, each obtained via (7.1)-(7.3), we can estimate the cdf $F_I(x) \equiv \mathbb{P}(I \leq x)$, $x \geq 0$, by the empirical cdf

$$\bar{F}_I(x) \equiv N^{-1} \sum_{i=1}^{N} I(I_i \leq x). \tag{7.4}$$

To estimate the p^{th} moment $\mathbb{E}[I^p]$, we can compute the sample mean, using

$$\bar{I}_N \equiv \rho R^{-1} \sum_{i=1}^{R} N^{-1} \sum_{i=1}^{N} I_i, \tag{7.5}$$

where R is the number of replications.

7.3. *A DTMC Numerical Algorithm.* If the traffic intensity ρ and the interarrival time $1 + c_a^2$ are integer multiples of a common $\delta > 0$, then the steps of the random walk are confined to a lattice subset of the real line and the possible values of the idle time lie in a finite subset. In particular, consider the alternative recursion

$$Z_{k+1} = Z_k + \rho N_k/\delta - (1 + c_a^2)/\delta, \quad k \geq 1, \quad Z_0 \equiv 0. \tag{7.6}$$

Clearly, each step in (7.1) is divided by δ in (7.6). Hence, $Y_k = \delta Z_k$, $k \geq 0$. However, now Z_k takes values in the integers. We assume that ρ and the interarrival time $1 + c_a^2$ are indeed integer multiples of a common δ and we use the largest δ with that property.

Thus, from (7.2) The number of customers served in that busy cycle, N_c, and the length of a busy cycle, C, are then

$$N_c = \inf\{k \geq 1 : Z_k \leq 0\} \quad \text{and} \quad C = N_c(1 + c_a^2)\delta. \tag{7.7}$$

The associated idle-time random variable is thus distributed as

$$I \stackrel{\mathrm{d}}{=} -\delta Z_{N_c}. \tag{7.8}$$

However, before hitting a nonpositive value, the random walk now must start in some nonnegative integer state. If the workload RW visits positive states, then it must start from a strictly positive integer, but we could have two idle times in a row. Then we could start in 0. Hence, we have

$$0 \leq -Z_{N_c} \leq \frac{1 + c_a^2 - \rho}{\delta} \quad \text{and} \quad 0 \leq I \leq 1 + c_a^2 - \rho. \tag{7.9}$$

Given the alternative recursion in (7.6), the random walk takes values in the integers, so we can calculate the distribution of I by calculating the absorption probabilities of a DTMC with integer state space. The absorption can take place on a finite subset of nonpositive integers. Specifically, the state space is the set $\mathcal{S} \equiv \{k : k \geq \rho/\delta - (1 + c_a^2)/\delta\}$ with absorbing states $\{k : -1 \geq k \geq \rho/\delta - (1 + c_a^2)/\delta\}$. We obtain a finite DTMC by truncating the state space at some level N; i.e., let the truncated state space be $\mathcal{S}^T \equiv \{k : \rho/\delta - (1 + c_a^2)/\delta \leq k \leq N\}$, let all transitions that initially go above N go instead to N, so that P is a legitimate DTMC.

As usual, let Q be the square submatrix of transition probabilities between transient states and let R be the submatrix of one-step transition probabilities from the transient states to the absorbing states. Let the fundamental matrix be $(I - Q)^{-1}$. Then the absorption probabilities are given by $B \equiv (I - Q)^{-1}R$. The first column of B corresponds to the absorption probabilities starting at state 0. We thus can use it to compute the moments $\mathbb{E}[I]$ and $\mathbb{E}[I^2]$.

7.4. *Numerical Experiments for the DTMC Algorithm.* To illustrate the DTMC numerical algorithm, we consider the example with $c_a^2 = 4$. First, Table 7 shows the results of the DTMC numerical algorithm for two values of ρ: 0.5 and 0.8. The required values of δ for these two cases are 1 and 0.2, respectively. We also show the performance for other (smaller) candidate δ, which satisfy the integer requirement, but make the state space larger.

TABLE 7
Performance of DTMC(N) with Different Truncation Levels N and δ

	$\rho = 0.8$		$\rho = 0.5$		
$N\backslash\delta$	0.2	0.1	0.5	0.25	0.1
1	14.831987	14.831987	3.456240	3.436333	3.436333
10	14.862050	14.842114	3.469846	3.473675	3.467565
100	14.913166	14.904170	3.470132	3.470132	3.470163
500	14.916936	14.916816	3.470132	3.470132	3.470132
1000	14.916937	14.916936	3.470132	3.470132	3.470132
2000	14.916937	14.916937	3.470132	3.470132	3.470132
5000	14.916937	14.916937	3.470132	3.470132	3.470132

Table 7 shows that both the truncation level N and the scale factor δ have an impact on $\mathbb{E}[W]$, but the algorithm converges with six decimal accuracy when N reaches 5E+03. The running time of algorithm depends on truncation level N. Constructing the $N \times N$ transition matrix requires computation of order $O((N + X)^2) = O(N^2)$, while computing the inverse matrix of Q. which is done by Gaussian elimination, requires $O(N^3)$. Hence, the overall complexity of the algorithm is $O(N^3)$.

To elaborate, Table 8 shows the performance of the DTMC algorithm as a function of N for other ρ. The appropriate δ is used in each case.

TABLE 8
Performance of DTMC Algorithm for Other Traffic Levels

$N\backslash\rho$	0.95	0.90	0.70	0.60	0.40	0.30
1E+00	74.512312	34.621172	8.372901	5.243412	2.289971	1.493015
1E+01	74.512312	34.696376	8.381077	5.267151	2.296621	1.498390
1E+02	74.568945	34.719782	8.434009	5.294671	2.304104	1.499233
5E+02	74.608460	34.719782	8.441300	5.294825	2.304105	1.499234
1E+03	74.616306	34.721369	8.441305	5.294825	2.304105	1.499234
2E+03	74.619898	34.721484	8.441305	5.294825	2.304105	1.499234
5E+03	74.620917	34.721484	8.441305	5.294825	2.304105	1.499234
1E+04	74.620917	34.721484	8.441305	5.294825	2.304105	1.499234

Finally, Table 9 shows the corresponding performance for $\rho = 0.99$, for which we need $\delta = 0.01$, leading to a larger number of possible idle times.

Given that the scale is 0.01, there are 102 possible idle time values, ranging from 0.00 to 4.01 in increments of 0.01, as indicated in (7.9). We report the results for different N.

TABLE 9
Performance of DTMC(N) for $\rho = 0.99$

$\delta \backslash N$	1E+02	5E+02	1E+03	2E+03	3E+03
0.01	394.420259	394.476457	394.496173	394.511729	394.518208
$\delta \backslash N$	5E+03	1E+04	2E+04	4E+04	6E+04
0.01	394.524273	394.529090	394.531611	394.533189	394.533189

Compared with performance of NB algorithm in this case, the DTMC algorithm is less efficient. The DTMC algorithm needs more than 1E+05 seconds CPU time for $N \geq 2E + 04$ to attain six decimal places accuracy for $\rho = 0.99$. In contrast, with only 7E+03 seconds cpu time, the NB can attains more than 15 decimal places accuracy. That advantage also holds for lower traffic intensities. For $\rho = 0.8$, NB only needs around 0.7 seconds CPU time for 15 decimal places accuracy while DTMC requires around 20 seconds cpu time with $N = 2000$.

8. Simulation Algorithms and Experiments. In this section we compare three different simulation algorithms for estimating the extremal mean steady-state waiting time $\mathbb{E}[W(F_0, G_{u^*})]$: (i) the standard Monte Carlo (MC) algorithm, (ii) the Minh-Sorli [12] (MS) algorithm and (iii) the method from §7.2 based on simulating a discrete-time random walk.

8.1. *The Simulation Algorithms.* We now describe the three simulation algorithms.

8.1.1. *Multiple Replications.* In order to estimate the overall statistical precision as well as to improve it, for each simulation experiment, we perform multiple (usually $20 - 40$) i.i.d. replications of the entire experiment. Thus, $\mathbb{E}[W]$ is estimated by the sample average

$$\bar{W}_R \equiv R^{-1} \sum_{i=1}^{R} \bar{W}_{[i]}, \tag{8.1}$$

where $\bar{W}_{[i]}$ is the estimate from the i^{th} replication and R is the number of replications.

By using multiple i.i.d. replications, we can construct confidence intervals in the standard way. In particular, the sample variance is

$$S^2 \equiv (1/(R-1)) \sum_{i=1}^{R} (\bar{W}_{[i]} - \bar{W}_R)^2, \tag{8.2}$$

so that the halfwidth of the confidence interval is $CIL = t^* S/\sqrt{R}$ where $t^* \equiv t(R)^*$ is the critical value of the Student statistical t-test with $R-1$ degrees of freedom. We use a 95% confidence interval, so $t(20)^* = 2.09$. To show the numerical and simulation methods accuracy, we compare the different computational methods with 95% confidence interval.

8.1.2. *The Standard Monte Carlo Algorithm.* The standard Monte-Carlo simulation method to estimate the mean steady-state waiting time in the $GI/GI/1$ queue exploits the Lindley recursion in (1.1). For each successive customer (indexed by n), we obtain a realization of the random variable W_n. The steady-state mean waiting time can be estimated by the sample average

$$\bar{W} \equiv \bar{W}(N) \equiv N^{-1} \sum_{n=1}^{N} W_n. \tag{8.3}$$

From (1.2), we see that the expected value of the estimate $\bar{W}(N)$ approaches the limit from below as N increases. Because the sequence $\{W_n : n \geq 0\}$ is a regenerative process, with empty times serving as regeneration points, we can apply the strong law of large numbers to deduce that the estimator is consistent as $N \to \infty$. As an alternative, we could use the regenerative approach in §IV.4 of [4].

In some cases, in order to reduce the estimation bias, within each replication we look at the long-run average after deleting an initial portion to allow the system to approach steady state. We exploit the two point distributions to simplify the event generation. In the simulation algorithm, the successive events are classified in three ways: (i) arrival is next, (ii) departure is next and (iii) next event occurs after given time T, where T is total simulation length.

The computational precision gradually improves as $N \to \infty$. Unfortunately, the algorithm is not efficient for $F_0/G_u/1$ with large M_s, primarily because the large service times are rare events, which cause significant problems; e.g., see §VI of [4] and §XIII.7 of [3]. Moreover, the standard simulation method is not efficient under heavy traffic levels because of its slow convergence; e.g., see [19].

8.1.3. *The Minh-Sorli* [12] *Simulation Algorithm.* In [12] another simulation algorithm was proposed to address the difficulty in heavy traffic. The idea is to exploit Theorem 6.1. In particular, we exploit the the discrete event simulation method to estimate the first two moments of the steady state idle period I; i.e., we exploit (6.1) and estimate $\phi(I)$ in (6.3). In the simulation algorithm, the successive events are classified in three ways: (i) arrival is next, (ii) departure is next and (iii) next event occurs after given time T, where T is total simulation length.

Thus, within each replication we estimate $\mathbb{E}[I]$ and $\mathbb{E}[I^2]$ and then apply Theorem 6.1 to obtain an associated estimate of $\mathbb{E}[W]$. We then compute confidence intervals for this alternative estimate of $\mathbb{E}[W]$ by performing multiple replications, as described in §8.1.1.

8.2. *Comparison of the Three Simulation Algorithms.* We now apply and compare our three simulation algorithms to estimate the mean steady-state waiting time in the extremal $F_0/G_{u^*}/1$ queue: (i) the standard Monte Carlo (MC) algorithm, (ii) the [12] (MS) algorithm and (iii) the method from §7.2 based on simulating a discrete-time random walk.

Estimates of $\mathbb{E}[W]$ for the $F_0/G_{u^*}/1$ model by the three algorithms are shown in Table 10. These are for the case $c_a^2 = c_s^2 = 4.0$ and $M_s = 1000$ for MC algorithm and $M_s = \infty$ for other two simulation algorithms. Results are reported for a range of traffic intensities ranging from $\rho = 0.1$ to $\rho = 0.99$.

We now describe the simulation parameters for each algorithm. The MC method had truncation level N =1E+07 in (8.3) and $R = 20$ i.i.d replications in (8.1.1). The MS method had total run length T =1E+06 again with $R = 20$ iid replications. (We used all idle periods that fall within that time interval.)

Table 10 shows the simulation estimates from all three approaches. Table 10 shows that the simulation methods are mutually confirming, but that the confidence intervals are quite different. The accuracy is ordered by $MS > RW > MC$ with MS being best.

TABLE 10
Comparison of Three Different Simulation Algorithms

simulation estimates of $\mathbb{E}[W(F_0/G_{u^*})]$ for $c_a^2 = c_s^2 = 4$						
ρ	MC UB	95% CI Length	MS UB	95% CI Length	RW UB	95% CI Length
0.10	0.422	5.08E-04	0.422	7.79E-05	0.422	9.28E-04
0.20	0.904	2.29E-03	0.904	1.30E-04	0.903	1.64E-03
0.30	1.484	4.44E-03	1.499	1.71E-04	1.498	1.47E-03
0.40	2.310	1.47E-02	2.304	1.90E-04	2.305	1.68E-03
0.50	3.472	2.15E-02	3.470	2.25E-04	3.472	2.00E-03
0.60	5.276	5.39E-02	5.294	2.43E-04	5.295	3.14E-03
0.70	8.381	7.80E-02	8.442	3.05E-04	8.442	2.62E-03
0.80	15.016	1.54E-01	14.917	3.22E-04	14.919	3.13E-03
0.90	34.525	4.60E-01	34.722	5.17E-04	34.720	1.95E-03
0.95	76.059	1.24E+00	74.621	7.11E-04	74.621	2.26E-03
0.98	193.206	3.07E+00	194.556	9.29E-04	194.558	2.75E-03
0.99	394.763	1.02E+01	394.532	1.45E-03	394.532	2.62E-03

8.2.1. *Simulation Efficiency.* To compare statistical efficiency and computational effectiveness, we consider the MC method with three different N, the RW method with three different N, and the MS method with three different total simulation time T. For each, 95% confidence intervals as a function of these parameters as well as the number R of replications numbers and the traffic intensity ρ are reported in Table 11.

TABLE 11
A Comparison of Three Simulation Methods

Confidence Interval Length for the MC method as a Function of N, R and ρ									
	$N = 5E+04$			$N = 1E+05$			$N = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	5.03E-01	2.60E+00	1.08E+01	3.73E-01	3.33E+00	1.09E+01	1.78E-01	4.88E-01	2.78E+00
30	4.85E-01	2.73E+00	1.11E+01	2.41E-01	1.25E+00	6.91E+00	1.42E-01	3.26E-01	2.90E+00
40	3.90E-01	1.48E+00	9.27E+00	2.66E-01	1.16E+00	4.60E+00	1.28E-01	2.85E-01	2.63E+00
50	3.95E-01	1.55E+00	6.34E+00	3.37E-01	1.04E+00	4.91E+00	1.07E-01	3.47E-01	1.79E+00
60	4.42E-01	1.10E+00	8.84E+00	2.61E-01	1.15E+00	5.14E+00	6.86E-02	3.41E-01	1.58E+00
70	3.32E-01	1.16E+00	7.32E+00	2.59E-01	8.35E-01	4.49E+00	8.67E-02	2.61E-01	1.52E+00
80	3.18E-01	1.29E+00	7.82E+00	2.78E-01	7.22E-01	5.18E+00	8.88E-02	2.78E-01	1.31E+00
90	3.87E-01	1.07E+00	6.35E+00	2.61E-01	9.79E-01	4.28E+00	7.33E-02	2.85E-01	1.29E+00
100	2.99E-01	1.04E+00	4.78E+00	2.14E-01	8.15E-01	3.76E+00	8.02E-02	2.22E-01	1.33E+00
Confidence Interval Length for the RW method with Number of Copies N									
	$N = 100$			$N = 500$			$N = 1000$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	1.77E-02	2.90E-02	2.27E-02	9.47E-03	1.06E-02	9.12E-03	8.13E-03	6.52E-03	7.43E-03
30	1.85E-02	1.83E-02	1.80E-02	6.78E-03	9.34E-03	7.82E-03	5.86E-03	5.07E-03	7.74E-03
40	1.51E-02	1.66E-02	1.73E-02	6.51E-03	8.11E-03	7.92E-03	5.25E-03	4.34E-03	6.14E-03
50	1.35E-02	1.49E-02	1.75E-02	5.84E-03	6.36E-03	7.06E-03	4.27E-03	3.97E-03	4.14E-03
60	1.21E-02	1.17E-02	1.39E-02	4.79E-03	6.02E-03	5.65E-03	3.49E-03	4.54E-03	4.24E-03
70	1.11E-02	1.30E-02	1.24E-02	4.81E-03	5.37E-03	5.84E-03	2.95E-03	3.44E-03	4.17E-03
80	1.14E-02	1.20E-02	1.11E-02	4.92E-03	3.90E-03	5.01E-03	3.08E-03	3.52E-03	3.78E-03
90	8.84E-03	9.94E-03	9.84E-03	4.18E-03	4.34E-03	4.62E-03	2.93E-03	3.15E-03	3.99E-03
100	8.30E-03	8.50E-03	1.09E-02	3.95E-03	4.22E-03	4.46E-03	2.95E-03	3.30E-03	3.42E-03
Confidence Interval Length for the MS method with Simulation Length T									
	$T = 1E+03$			$T = 1E+04$			$T = 1E+05$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	1.88E-02	1.91E-02	2.42E-02	5.51E-03	7.87E-03	9.33E-03	1.34E-03	2.01E-03	3.16E-03
30	1.31E-02	1.47E-02	3.78E-02	4.50E-03	5.27E-03	9.97E-03	9.59E-04	1.36E-03	2.43E-03
40	1.01E-02	1.56E-02	2.67E-02	4.04E-03	4.78E-03	8.65E-03	1.19E-03	1.56E-03	2.94E-03
50	1.04E-02	1.39E-02	2.25E-02	3.35E-03	4.02E-03	7.47E-03	8.93E-04	1.46E-03	2.11E-03
60	9.72E-03	1.21E-02	2.39E-02	2.60E-03	3.51E-03	6.65E-03	7.58E-04	1.03E-03	1.91E-03
70	9.32E-03	8.66E-03	1.87E-02	2.51E-03	3.74E-03	5.96E-03	8.77E-04	1.16E-03	1.99E-03
80	8.55E-03	9.71E-03	1.78E-02	2.07E-03	3.31E-03	7.06E-03	8.62E-04	1.16E-03	1.70E-03
90	6.85E-03	8.56E-03	1.59E-02	2.22E-03	3.30E-03	5.74E-03	7.13E-04	9.58E-04	1.57E-03
100	7.74E-03	8.46E-03	1.81E-02	2.14E-03	3.04E-03	4.72E-03	7.49E-04	8.71E-04	1.37E-03

The MS and RW methods are based on sample means from i.i.d. samples and thus are unbiased estimators, but that is not the case for MC. So the bias is also a concern, especially for high ρ. Thus, the MC method is even worse than shown. To illustrate the problem, we compare the RW and MC algorithms for $\rho = 0.99$ in Table 12. Table 12 shows the large error for smaller N with MC, but no problem at all with RW.

TABLE 12
A Comparison between MC and RW Simulation for $\rho = 0.99$

	$N = 1E+02$	$N = 1E+02$	$N = 5E+02$	$N = 5E+02$	$N = 1E+03$	$N = 1E+03$
$R = 100$	$\mathbb{E}[W]$	95% CIL	$\mathbb{E}[W]$	95% CIL	$\mathbb{E}[W]$	95% CIL
RW	394.533	1.02E-02	394.530	4.57E-03	394.535	3.29E-03
	$N = 5E+04$	$N = 5E+04$	$N = 1E+05$	$N = 1E+05$	$N = 1E+06$	$N = 1E+06$
$R = 100$	$\mathbb{E}[W]$	95% CIL	$\mathbb{E}[W]$	95% CIL	$\mathbb{E}[W]$	95% CIL
MC	182.41	2.43E+01	261.62	3.30E+01	385.48	3.34E+01

After comparing the computational outcomes from these three tables,

we see that the MS algorithm clearly is more efficient than the other two simulation algorithms. To elaborate, we describe the computational effort. With 100 seconds of CPU time and 100 iid replications, the MS method can reach 1E-04 95% confidence interval length for most of the traffic levels, while the MC can only have 1E-03 confidence interval length.

Expressed differently, in order to achieve 1E-03 or 1E-02 confidence interval length for all traffic levels, the MS method needs at most needs CPU computational time less than 1 second, but RW needs several seconds. The MC method is the worst method which has bad performance in computational cost and accuracy typically for heavy traffic. Even though it takes more than 200 seconds CPU time with 100 replications and N=1E+06 copies, the confidence interval length can still be large than 1 for some heavy traffic levels.

Finally, the MC and MS methods are far easier to generalize. The MC method applies to many models, while the MS method applies to any $GI/GI/1$ queue, but the RW method depends on the detailed special structure. Hence, there exist more strict requirements to implement the RW method.

8.3. *Simulation Comparisons for Three Related Models.* In order to better understand the c0mputational issues provided by the extremal $F_0/G_{u^*}/1$ model, we now compare the MC and MS algorithms on three different models: (i) the $F_0/G_u/1$ with $M_s = 1000$, (ii) the $F_0/D/1$ model (avoiding the rare large service time) and (iii) the reduced $D(1/p)/RS(D(\rho), p)/1$ model obtained from the model reductions.

8.3.1. *A Monte Carlo Simulation Comparison for Three Queues.* We now compare MC simulation performance for three queues $F_0/G_u/1$ with $M_s = 10^3$, $F_0/D/1$ and $D/RS(\rho, p)/1$ for traffic level $\rho = 0.5, 0.7, 0.9$ and report the confidence interval length based on statistical T test.

TABLE 13
A Comparison of Monte-Carlo simulation for Two Queues

Confidence Interval Length for MC for $F_0/G_u/1$ with $M_s = 1000$									
	$N = 5E+04$			$N = 1E+05$			$N = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	5.03E-01	2.60E+00	1.08E+01	3.73E-01	3.33E+00	1.09E+01	1.78E-01	4.88E-01	2.78E+00
40	3.90E-01	1.48E+00	9.27E+00	2.66E-01	1.16E+00	4.60E+00	1.28E-01	2.85E-01	2.63E+00
60	4.42E-01	1.10E+00	8.84E+00	2.61E-01	1.15E+00	5.14E+00	6.86E-02	3.41E-01	1.58E+00
80	3.18E-01	1.29E+00	7.82E+00	2.78E-01	7.22E-01	5.18E+00	8.88E-02	2.78E-01	1.31E+00
100	2.99E-01	1.04E+00	4.78E+00	2.14E-01	8.15E-01	3.76E+00	8.02E-02	2.22E-01	1.33E+00
Confidence Interval Length for MC for $F_0/D/1$									
	$N = 5E+04$			$N = 1E+05$			$N = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	4.60E-03	4.99E-03	1.40E-02	1.72E-03	1.54E-03	3.39E-03	4.25E-04	7.84E-04	1.23E-03
40	3.41E-03	4.31E-03	7.89E-03	1.18E-03	1.36E-03	2.57E-03	3.16E-04	4.25E-04	8.54E-04
60	2.94E-03	3.77E-03	6.14E-03	8.50E-04	1.30E-03	2.22E-03	2.93E-04	3.50E-04	6.49E-04
80	2.63E-03	3.30E-03	5.49E-03	8.19E-04	1.01E-03	1.83E-03	2.56E-04	2.85E-04	4.96E-04
100	2.43E-03	2.89E-03	5.31E-03	8.18E-04	9.07E-04	1.40E-03	1.87E-04	2.86E-04	4.45E-04
Confidence Interval Length of MC for $D(1/p)/RS(D(\rho), p)/1$									
	$N = 5E+04$			$N = 1E+05$			$N = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	6.19E-03	3.40E-02	4.76E-01	4.61E-03	2.08E-02	3.23E-01	1.61E-03	7.61E-03	8.19E-02
40	3.29E-03	2.66E-02	2.92E-01	2.61E-03	2.00E-02	2.19E-01	1.04E-03	6.46E-03	7.13E-02
60	3.03E-03	1.79E-02	2.80E-01	2.07E-03	1.16E-02	1.68E-01	7.27E-04	4.79E-03	6.03E-02
80	2.62E-03	1.89E-02	2.10E-01	2.04E-03	1.19E-02	1.47E-01	5.75E-04	3.67E-03	4.63E-02
100	2.82E-03	1.57E-02	1.90E-01	1.63E-03	9.84E-03	1.23E-01	6.19E-04	3.14E-03	4.83E-02

As expected, Table 13 shows that the model reduction makes the Monte-Carlo simulation more efficient and accurate. Typically, the simulation is most accurate for $F_0/D/1$.

8.3.2. *A Minh-Sorli Simulation Comparison for Three Queues..* We have shown MS method has the same performance for the two queues $F_0/D/1$ and $F_0/G_u/1$ as $M_s \to \infty$ in §4. So we compare the simulation performance for $F_0/G_u/1$ with given $M_s = 1000$, $F_0/D/1$ and the queue $D/RS(\rho, p)/1$.

TABLE 14
A Comparison of Minh-Sorli simulation for Three Queues

Confidence Interval Length of MS for $F_0/G_u/1$									
	$T = 5E+04$			$T = 1E+05$			$T = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	1.88E-02	1.91E-02	2.42E-02	5.51E-03	7.87E-03	9.33E-03	1.34E-03	2.01E-03	3.16E-03
40	1.01E-02	1.56E-02	2.67E-02	4.04E-03	4.78E-03	8.65E-03	1.19E-03	1.56E-03	2.94E-03
60	9.72E-03	1.21E-02	2.39E-02	2.60E-03	3.51E-03	6.65E-03	7.58E-04	1.03E-03	1.91E-03
80	8.55E-03	9.71E-03	1.78E-02	2.07E-03	3.31E-03	7.06E-03	8.62E-04	1.16E-03	1.70E-03
100	7.74E-03	8.46E-03	1.81E-02	2.14E-03	3.04E-03	4.72E-03	7.49E-04	8.71E-04	1.37E-03
Confidence Interval Length of MS for $F_0/D/1$									
	$T = 5E+04$			$T = 1E+05$			$T = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	4.07E-03	5.04E-03	1.13E-02	3.61E-03	3.96E-03	8.32E-03	1.05E-03	1.33E-03	2.86E-03
40	3.28E-03	4.12E-03	6.79E-03	2.20E-03	2.23E-03	4.18E-03	6.46E-04	8.24E-04	1.72E-03
60	2.57E-03	2.77E-03	6.67E-03	1.75E-03	2.91E-03	3.66E-03	4.85E-04	6.94E-04	1.49E-03
80	2.22E-03	3.05E-03	4.51E-03	1.59E-03	2.04E-03	3.44E-03	5.04E-04	6.27E-04	1.06E-03
100	1.65E-03	2.63E-03	4.27E-03	1.32E-03	1.51E-03	3.49E-03	4.43E-04	5.28E-04	9.82E-04
Confidence Interval Length of MS for $D(1/p)/RS(D(\rho), p)/1$									
	$T = 5E+04$			$T = 1E+05$			$T = 1E+06$		
$R\backslash\rho$	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
20	4.60E-03	5.74E-03	1.10E-02	2.43E-03	4.16E-03	9.07E-03	9.40E-04	9.97E-04	2.54E-03
40	3.82E-03	3.26E-03	6.97E-03	2.43E-03	3.22E-03	5.97E-03	7.31E-04	9.14E-04	1.88E-03
60	2.48E-03	3.33E-03	6.66E-03	1.77E-03	2.34E-03	4.26E-03	5.40E-04	6.64E-04	1.37E-03
80	1.89E-03	2.48E-03	4.68E-03	1.68E-03	2.06E-03	3.11E-03	5.18E-04	6.36E-04	1.16E-03
100	1.89E-03	2.56E-03	3.95E-03	1.16E-03	1.51E-03	3.20E-03	4.33E-04	5.36E-04	9.18E-04

The Minh-Sorli algorithm for all queues have the almost same simulation accuracy, typically $F_0/D/1$ and $D/RS(\rho, p)/1$ are slightly better than $F_0/G_u/1$. Regarding the computational effort, the cpu time is around $20-100$ seconds for $F_0/D/1$ while that is around $50-300$ seconds for $D/RS(\rho, p)/1$ when R increases from 20 to 100. So The model reduction makes the Minh-Sorli algorithm more efficient.

Tables 13 and 14 show that the inter-arrival-time and service-time model reductions both make the algorithms more accurate and efficient, but the service-time reduction is slightly better. Moreover, the Minh-Sorli simulation outperforms Monte-Carlo simulation for any of the three models.

8.3.3. *The Idle-Time Distribution in Two Queues.* We apply the Minh-Sorli [12] simulation algorithm to compare the first two moments of steady-state idle time for the extremal queue $F_0/G_{u^*}/1$ queue and the $M/M/1$ queue.

For the $M/M/1$ model with $\lambda = 1$, it is well known that both I and I_e are exponential with mean 1 for all ρ, so that $\mathbb{E}[I] = 1$, $\mathbb{E}[I^2] = 2$ and $\mathbb{E}[I_e] = 1$ for all ρ. Nevertheless, as an independent check, we apply the MS algorithm to both the $M/M/1$ and $F_0/G_{u^*}/1$ models. The results are shown in Table 15.

Figure 1 shows an estimate of the steady-state idle-time distribution by MS. To get good precision, we increase T to $T = 5E+09$ under $\rho = 0.99$. We remark that this is also the steady-state idle-time distribution for model

TABLE 15
A Comparison of the idle-time Distribution in the $F_0/G_{u^*}/1$ and $M/M/1$ queues, using the Minh-Sorli [12] algorithm with $T = 1E + 06$

		$\rho = 0.8$			$\rho = 0.99$		
	R	$\mathbb{E}[I]$	$\mathbb{E}[I^2]$	$\mathbb{E}[I_e]$	$\mathbb{E}[I]$	$\mathbb{E}[I^2]$	$\mathbb{E}[I_e]$
$F_0/G_{u^*}/1$	20	2.453	7.766	1.583	2.111	6.298	1.492
	40	2.452	7.765	1.583	2.114	6.307	1.492
	60	2.452	7.763	1.583	2.114	6.304	1.491
	80	2.451	7.760	1.583	2.114	6.309	1.492
	100	2.451	7.760	1.583	2.113	6.306	1.492
		$\rho = 0.8$			$\rho = 0.99$		
	R	$\mathbb{E}[I]$	$\mathbb{E}[I^2]$	$\mathbb{E}[I_e]$	$\mathbb{E}[I]$	$\mathbb{E}[I^2]$	$\mathbb{E}[I_e]$
$M/M/1$	20	1.000	1.999	1.000	1.000	2.003	1.001
	40	0.999	1.997	0.999	0.999	1.994	0.997
	60	1.000	1.999	1.000	1.002	2.002	0.999
	80	1.000	1.999	1.000	1.001	2.005	1.001
	100	1.000	2.001	1.000	1.000	2.002	1.001

$F_0/D/1$.

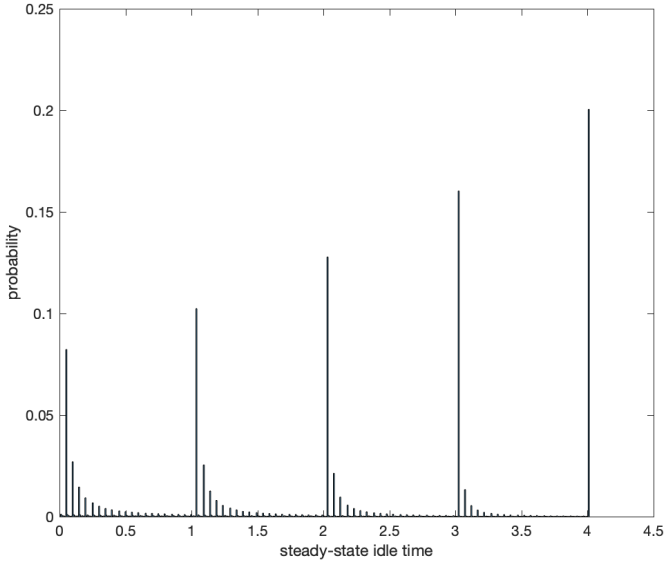

FIG 1. *Simulation estimates of the steady-state idle-time distribution in the $F_0/G_{u^*}/1$ model under traffic level $\rho = 0.99$.*

9. Conclusions. In this paper we developed numerical and simulation algorithms to compute the mean steady waiting time $\mathbb{E}[W]$ in the extremal $GI/GI/1$ queue given the first two moments of the interarrival-time and

service-time distributions, as specified by the parameter vector $(1, c_a^2, \rho, c_s^2)$. In [5] we showed that $\mathbb{E}[W]$ is attained (asymptotically) in the $F_0/G_{u^*}/1$ model, involving two-point distributions. In §2 we present evidence that the tight upper bound provides a significant improvement over previous upper bounds.

Our algorithms are based on three different convenient alternative representations for the mean waiting time $\mathbb{E}[W]$ in the $F_0/G_{u^*}/1$ extremal model. In §3 and §4, we showed that it suffices to calculate $\mathbb{E}[W]$ in the $D(1/p)/RS(D(\rho), p)/1$ model, where $p = 1/(1 + c_a^2)$ and the service time is a geometric random sum of deterministic values taking the value ρ.

In §5 we developed effective numerical algorithms to compute the mean steady-state waiting time $\mathbb{E}[W(D(1/p)/RS(D(\rho), p)/1]$ using recursive algorithms for the negative binomial distribution. We also conducted experiments showing that they are effective. We exposed and resolved an underflow problem that can arise in heavy traffic.

In §6 we showed that it also suffices the compute the first two moments of the steady-state idle-time distribution in the $D(1/p)/RS(D(\rho), p)/1$ model. Theorem 6.2 shows that the idle time is better behaved than the waiting time as the extremal service mass increases. In §7 we showed that effective numerical and simulation algorithms can be developed based on this approach as well, but so far this approach does not seem better than the NB algorithm in §5.

In §8 we studied three possible simulation algorithms for estimating $\mathbb{E}[W]$ in the $F_0/G_{u^*}/1$ model: the standard monte Carlo simulation (MC) and two methods exploiting the idle-time representation: the Minh-Sorli [12] algorithm and a new algorithm based on a discrete time random walk (RW). We showed that both MS and RW provide significant improvement over MC, but that MS tends to be best.

Overall, we found that, first, the reductions are powerful for simplifying the algorithms and, second, that the refined negative-binomial numerical algorithm in §5 and the Minh-Sorli [12] simulation algorithm in §8 are most effective for computing $\mathbb{E}[W(D(1/p)/RS(D(\rho), p)/1]$.

10. Appendix. We now present additional results to supplement the main paper. Tables 16 and 17 are analogs of Tables 1 and 2 for the mixed cases $c_a^2 = 4.0, c_s^2 = 0.5$. and $c_a^2 = 0.5, c_s^2 = 4.0$.

TABLE 16
A comparison of the unscaled bounds and approximations for the steady-state mean $\mathbb{E}[W]$ *as a function of* ρ *for the case* $c_a^2 = 4.0$ *and* $c_s^2 = 0.5$

ρ	Tight LB (1.9)	HTA (2.1)	Tight UB	UB Approx (1.8)	δ	MRE	Daley (1.6)	Kingman (1.5)
0.10	0.00	0.025	0.403	0.403	0.000	0.00%	0.425	2.23
0.15	0.00	0.060	0.607	0.607	0.001	0.06%	0.660	2.36
0.20	0.00	0.113	0.816	0.818	0.007	0.21%	0.913	2.51
0.25	0.00	0.188	1.04	1.04	0.020	0.45%	1.19	2.69
0.30	0.00	0.289	1.27	1.28	0.041	0.76%	1.49	2.89
0.35	0.00	0.424	1.54	1.55	0.070	1.10%	1.82	3.12
0.40	0.00	0.600	1.83	1.86	0.107	1.31%	2.20	3.40
0.45	0.00	0.828	2.18	2.21	0.152	1.63%	2.63	3.73
0.50	0.00	1.13	2.60	2.64	0.203	1.51%	3.13	4.13
0.55	0.00	1.51	3.08	3.14	0.261	1.89%	3.71	4.61
0.60	0.00	2.03	3.71	3.78	0.324	1.79%	4.43	5.23
0.65	0.00	2.72	4.51	4.59	0.393	1.62%	5.32	6.02
0.70	0.00	3.68	5.56	5.66	0.467	1.74%	6.48	7.08
0.75	0.00	5.06	7.07	7.17	0.546	1.39%	8.06	8.56
0.80	0.00	7.20	9.29	9.42	0.629	1.31%	10.40	10.80
0.85	0.28	10.84	13.04	13.17	0.716	0.93%	14.24	14.54
0.90	1.08	18.23	20.53	20.67	0.807	0.68%	21.83	22.03
0.95	3.54	40.61	43.00	43.17	0.902	0.39%	44.41	44.51
0.98	11.02	108.0	110.5	110.7	0.960	0.17%	112.0	112.0
0.99	23.51	220.5	223.0	223.2	0.980	0.09%	224.5	224.5

TABLE 17
A comparison of the unscaled bounds and approximations for the steady-state mean $\mathbb{E}[W]$ *as a function of* ρ *for the case* $c_a^2 = 0.5$ *and* $c_s^2 = 4.0$

ρ	Tight LB (1.9)	HTA (2.1)	Tight UB	UB Approx (1.8)	δ	MRE	Daley (1.6)	Kingman (1.5)
0.10	0.00	0.025	0.072	0.072	0.000	0.03%	0.075	0.300
0.15	0.00	0.060	0.128	0.128	0.001	0.03%	0.135	0.347
0.20	0.00	0.113	0.200	0.201	0.007	0.30%	0.213	0.413
0.25	0.00	0.188	0.292	0.294	0.020	0.68%	0.313	0.500
0.30	0.00	0.289	0.409	0.414	0.041	1.07%	0.439	0.614
0.35	0.00	0.424	0.558	0.565	0.070	1.32%	0.599	0.762
0.40	0.00	0.600	0.746	0.757	0.107	1.48%	0.800	0.950
0.45	0.011	0.828	0.986	1.00	0.152	1.58%	1.05	1.19
0.50	0.250	1.13	1.29	1.31	0.203	1.91%	1.38	1.50
0.55	0.569	1.51	1.69	1.72	0.261	1.45%	1.79	1.90
0.60	1.000	2.03	2.21	2.24	0.324	1.40%	2.33	2.43
0.65	1.589	2.72	2.91	2.95	0.393	1.26%	3.04	3.13
0.70	2.427	3.68	3.88	3.92	0.467	1.23%	4.03	4.10
0.75	3.63	5.06	5.25	5.33	0.546	1.41%	5.44	5.50
0.80	5.50	7.20	7.42	7.48	0.629	0.74%	7.60	7.65
0.85	8.71	10.8	11.18	11.13	0.716	0.48%	11.3	11.3
0.90	15.3	18.2	18.47	18.53	0.807	0.32%	18.7	18.7
0.95	35.1	40.6	40.87	40.93	0.902	0.15%	41.1	41.1
0.98	95.1	108.0	108.3	108.4	0.960	0.06%	108.5	108.5
0.99	195.0	220.5	220.8	220.9	0.980	0.03%	221.0	221.0

Acknowledgement. Research support was received from NSF (CMMI 1634133).

REFERENCES

[1] J. Abate, G. L. Choudhury, and W. Whitt. Calculation of the GI/G/1 steady-state waiting-time distribution and its cumulants from Pollaczek's formula. *Archiv fur Elektronik und bertragungstechnik*, 47(5/6):311–321, 1993.

[2] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10:5–88, 1992.

[3] S. Asmussen. *Applied Probability and Queues*. Springer, New York, second edition, 2003.

[4] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, New York, second edition, 2007.

[5] Y. Chen and W. Whitt. Extremal $GI/GI/1$ queues given two moments. Columbia University, http://www.columbia.edu/∼ww2040/allpapers.html, 2018.

[6] K. L. Chung. *A Course in Probability Theory*. Academic Press, New York, third edition, 2001.

[7] D. J. Daley. Inequalities for moments of tails of random variables, with queueing applications. *Zeitschrift fur Wahrscheinlichkeitsetheorie Verw. Gebiete*, 41:139–143, 1977.

[8] D. J. Daley, A. Ya. Kreinin, and C.D. Trengove. Inequalities concerning the waiting-time in single-server queues: a survey. In U. N. Bhat and I. V. Basawa, editors, *Queueing and Related Models*, pages 177–223. Clarendon Press, 1992.

[9] S. Halfin. Batch delays versus customer delays. *Bell Laboratories Technical Journal*, 62(7):2011–2015, 1983.

[10] J. F. C. Kingman. Inequalities for the queue $GI/G/1$. *Biometrika*, 49(3/4):315–324, 1962.

[11] K. T. Marshall. Some inequalities in queueing. *Operations Research*, 16(3):651–668, 1968.

[12] D. L. Minh and R. M. Sorli. Simulating the $GI/G/1$ queue in heavy traffic. *Operations Research*, 31(5):966–971, 1983.

[13] S. M. Ross. *Stochastic Processes*. Wiley, New York, second edition, 1996.

[14] S. M. Ross. *Introduction to Probability Models*. Academic Press, New York, eleventh edition, 2014.

[15] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. John Wiley and Sons, New York, 1983. Translated and edited from 1977 German Edition by D. J. Daley.

[16] D. Stoyan and H. Stoyan. Inequalities for the mean waiting time in single-line queueing systems. *Engineering Cybernetics*, 12(6):79–81, 1974.

[17] W. Whitt. Comparing batch delays and customer delays. *Bell Laboratories Technical Journal*, 62(7):2001–2009, 1983.

[18] W. Whitt. On approximations for queues, I. *AT&T Bell Laboratories Technical Journal*, 63(1):115–137, 1984.

[19] W. Whitt. Planning queueing simulations. *Management Science*, 35(11):1341–1366, 1989.

[20] W. Whitt. Engineering solution of a basic call-center model. *Management Sci.*, 51:221–235, 2005.

[21] R. W. Wolff and C. Wang. Idle period approximations and bounds for the $GI/G/1$ queue. *Advances in Applied Probability*, 35(3):773–792, 2003.

Department of IEOR,
S. W. Mudd Building,
500 West 120th Street,
New York, NY 10027-6699
E-mail: yc3107@columbia.edu
ww2040@columbia.edu