Online Supplement to Algorithms for the Upper Bound Mean Waiting Time in the $GI/GI/1$ Queue

Yan Chen
Industrial Engineering and Operations Research, Columbia University, yc3107@columbia.edu
Ward Whitt
Industrial Engineering and Operations Research, Columbia University, ww2040@columbia.edu

From the main paper: It has long been conjectured that 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 is attained asymptotically by two-point distributions. The two-point distribution for the interarrival time has one mass point at 0, but the service-time distribution involves a limit; there is one mass point at a high value, but that upper mass point must increase to infinity while the probability on that point must decrease to 0 appropriately. In this paper we develop effective numerical and simulation algorithms to compute the value of this conjectured tight bound. The algorithms are aided by reductions of the special queues with extremal intarrival-time and extremal service-time distributions to $D/GI/1$ and $GI/D/1$ models. Combining these reductions yields an overall representation in terms of a $D/RS(D)/1$ discrete-time model involving a geometric random sum of deterministic random variables (the $RS(D)$), where the two deterministic random variables in the model may have different values, so that the extremal steady-state waiting time need not have a lattice distribution. Efficient computational methods are developed. The computational results show that the conjectured tight upper bound offers a significant improvement over established bounds.

Key words: the single-server queue, bounds for the mean waiting time, extremal queues, stochastic simulation, two-point distributions
History: April 30, 2019; revised November 7, 2019

1. Introduction

In this supplement to the main paper Chen and Whitt (2019) we present additional supporting material. We start in §2 by providing brief overview of Chen and Whitt (2018, 2019) by comparing the classic bounds §2.2 of the main paper to the tight bounds and the heavy-traffic approximation in equation (9) in the main paper. In §3 we elaborate on the random walk representation for the steady-state idle time I and discuss both a numerical

algorithm and a simulation algorithm based on it. In §4 we elaborate on the simulation algorithms and in §5 we describe the results of additional simulation experiments.

2. A Comparison of Different Bounds and Approximations

To show that the tight UB $\mathbb{E}[W(F_0, G_{u^*})]$ defined in (11) of the main paper provides a significant improvement, we compared the estimates of the tight UB in the $GI/GI/1$ model with given first two moments associated with $(c_a^2, c_s^2) = (4.0, 4.0)$, as estimated by the Minh and Sorli (1983) simulation algorithm, to other bounds and approximations in Table 1 of the main paper. Comparisons for the other cases $(c_a^2, c_s^2) = (0.5, 0.5)$, $(4.0, 0.5)$ and $(0.5, 4.0)$ appear in Tables 1, 2 and 3 here.

We refer to the equations in §2.2 of the main paper. Our algorithms compute the "Tight UB" in these tables, given in (15) of the main paper, while the LB formula is (10), the new UB established in Theorem 2 of the main paper is (17), the Daley (1977) bound is (7) and the Kingman (1962) bound is (6). The common heavy-traffic approximation (HTA) is (9) in the main paper, i.e.,

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

The MRE is the maximum relative error between the new bound in (17) and the tight UB. The maximum value of the MRE for each of the cases $(c_a^2, c_s^2) = (4.0, 4.0)$, $(0.5, 0.5)$, $(4.0, 0.5)$ and $(0.5, 4.0)$ were, respectively, 1.2%, 5.7%, 1.5% and 1.9%. In all cases these occur at approximately $\rho = 0.5$.

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 $c_a^2 = c_s^2 = 0.5$. (Equation numbers given in the main paper.)**

ρ	Tight LB (10)	HTA (9)	Tight UB (15)	new UB (17)	δ (18)	MRE	Daley (7)	Kingman (6)
0.10	0.000	0.006	0.053	0.053	0.000	0.00%	0.056	0.281
0.15	0.000	0.013	0.082	0.082	0.001	0.11%	0.088	0.301
0.20	0.000	0.025	0.113	0.113	0.007	0.54%	0.125	0.325
0.25	0.000	0.042	0.146	0.148	0.020	1.35%	0.167	0.354
0.30	0.000	0.064	0.184	0.189	0.041	2.36%	0.214	0.389
0.35	0.000	0.094	0.228	0.235	0.070	3.16%	0.269	0.432
0.40	0.000	0.133	0.280	0.291	0.107	3.82%	0.333	0.483
0.45	0.000	0.184	0.342	0.357	0.152	4.43%	0.409	0.547
0.50	0.000	0.250	0.414	0.439	0.203	5.72%	0.500	0.625
0.55	0.000	0.336	0.515	0.540	0.261	4.62%	0.611	0.724
0.60	0.000	0.450	0.637	0.669	0.324	4.71%	0.750	0.850
0.65	0.000	0.604	0.800	0.837	0.393	4.45%	0.929	1.016
0.70	0.058	0.817	1.017	1.065	0.467	4.53%	1.167	1.242
0.75	0.188	1.125	1.312	1.388	0.546	5.42%	1.500	1.563
0.80	0.400	1.600	1.822	1.877	0.629	2.95%	2.000	2.050
0.85	0.779	2.408	2.646	2.700	0.716	1.99%	2.833	2.871
0.90	1.575	4.050	4.295	4.355	0.807	1.38%	4.500	4.525
0.95	4.037	9.025	9.284	9.344	0.902	0.65%	9.500	9.512
0.98	11.515	24.010	24.271	24.338	0.960	0.27%	24.500	24.505
0.99	24.008	49.005	49.265	49.336	0.980	0.14%	49.500	49.503

Table 2 **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 $c_s^2 = 0.5$ (Equation numbers given in the main paper.)**

ρ	Tight LB (10)	HTA (9)	Tight UB (15)	new UB (17)	δ (18)	MRE	Daley (7)	Kingman (6)
0.10	0.000	0.025	0.403	0.403	0.000	0.00%	0.425	2.225
0.15	0.000	0.060	0.607	0.607	0.001	0.05%	0.660	2.360
0.20	0.000	0.113	0.816	0.818	0.007	0.21%	0.913	2.513
0.25	0.000	0.188	1.036	1.041	0.020	0.45%	1.188	2.688
0.30	0.000	0.289	1.274	1.283	0.041	0.71%	1.489	2.889
0.35	0.000	0.424	1.538	1.553	0.070	0.96%	1.824	3.124
0.40	0.000	0.600	1.837	1.859	0.107	1.16%	2.200	3.400
0.45	0.000	0.828	2.184	2.214	0.152	1.35%	2.628	3.728
0.50	0.000	1.125	2.595	2.635	0.203	1.51%	3.125	4.125
0.55	0.000	1.513	3.096	3.144	0.261	1.53%	3.713	4.613
0.60	0.000	2.025	3.720	3.777	0.324	1.50%	4.425	5.225
0.65	0.000	2.716	4.519	4.586	0.393	1.45%	5.316	6.016
0.70	0.058	3.675	5.583	5.662	0.467	1.39%	6.475	7.075
0.75	0.188	5.063	7.077	7.165	0.546	1.23%	8.063	8.563
0.80	0.400	7.200	9.317	9.417	0.629	1.06%	10.400	10.800
0.85	0.779	10.838	13.055	13.168	0.716	0.86%	14.238	14.538
0.90	1.575	18.225	20.546	20.668	0.807	0.59%	21.825	22.025
0.95	4.037	40.613	43.033	43.168	0.902	0.31%	44.413	44.513
0.98	11.515	108.045	110.479	110.667	0.960	0.17%	111.965	112.005
0.99	24.008	220.523	222.971	223.167	0.980	0.09%	224.483	224.503

Table 3 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 $c_s^2 = 4.0$. (Equation numbers given in the main paper.)

ρ	Tight LB (10)	HTA (9)	Tight UB (15)	new UB (17)	δ (18)	MRE	Daley (7)	Kingman (6)
0.10	0.000	0.025	0.072	0.072	0.000	0.00%	0.075	0.300
0.15	0.000	0.060	0.128	0.128	0.001	0.07%	0.135	0.347
0.20	0.000	0.113	0.200	0.201	0.007	0.30%	0.213	0.413
0.25	0.042	0.188	0.292	0.294	0.020	0.68%	0.313	0.500
0.30	0.107	0.289	0.409	0.414	0.041	1.08%	0.439	0.614
0.35	0.202	0.424	0.558	0.565	0.070	1.32%	0.599	0.762
0.40	0.333	0.600	0.746	0.757	0.107	1.47%	0.800	0.950
0.45	0.511	0.828	0.986	1.002	0.152	1.58%	1.053	1.191
0.50	0.750	1.125	1.289	1.314	0.203	1.91%	1.375	1.500
0.55	1.069	1.513	1.692	1.716	0.261	1.45%	1.788	1.900
0.60	1.500	2.025	2.212	2.244	0.324	1.40%	2.325	2.425
0.65	2.089	2.716	2.913	2.950	0.393	1.26%	3.041	3.129
0.70	2.917	3.675	3.875	3.923	0.467	1.23%	4.025	4.100
0.75	4.125	5.063	5.250	5.325	0.546	1.41%	5.438	5.500
0.80	6.000	7.200	7.422	7.477	0.629	0.74%	7.600	7.650
0.85	9.208	10.838	11.075	11.129	0.716	0.48%	11.263	11.300
0.90	15.750	18.225	18.470	18.530	0.807	0.32%	18.675	18.700
0.95	35.625	40.613	40.871	40.932	0.902	0.15%	41.088	41.100
0.98	95.550	108.045	108.307	108.373	0.960	0.06%	108.535	108.540
0.99	195.525	220.523	220.783	220.853	0.980	0.03%	221.018	221.020

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 Kingman (1962) bound in (6) to the Daley (1977) bound in (7) to the new UB formula in (17). We also see that the tight UB in (15) is very well approximated by the UB formula in (17), but it requires calculating the root of an equation.

In closing this section, we emphasize that it remains to prove: (i) that (17) 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 2 of the main paper proves (i) under the assumption that (ii) is correct. Nevertheless, we have provided strong numerical evidence that the UB is $\mathbb{E}[W(F_0, G_{u^*})]$ in (11) and Theorem 1 of the main paper, is the tight UB. If that can be accepted, then the algorithms in the main paper provide effective ways to calculate the tight UB and formula (17) serves as an excellent approximation.

3. Computing the Distribution and Moments of the Idle Time

Theorem 7 of the main paper 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 $D(1/p)/RS(D(\rho),p)/1$ 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 $D(1/p)/RS(D(\rho),p)/1$ model based on a random walk representation.

3.1. A Random Walk Absorption Representation of the Idle-Time

We first review the random walk representation for the idle time I in the reduced model $D(1/p)/RS(D,p)/1$ model given in §8.2.1 of the main paper. Then we discuss a numerical algorithm. 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,

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

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

$$N_c = \inf\{k \geq 1 : Y_k \leq 0\} \quad \text{and} \quad C = N_c(1 + c_a^2). \tag{3}$$

The associated idle-time random variable is distributed as

$$I \stackrel{\text{d}}{=} -Y_{N_c}, \quad \text{so that} \quad 0 \leq I \leq c_a^2 + 1 - \rho. \tag{4}$$

3.2. An Idle-Time Simulation Algorithm

Given N i.i.d. copies of I, each obtained via (2)-(4), we can estimate the cdf $F_I(x) \equiv \mathbb{P}(I \leq x)$, $x \geq 0$, by the empirical cdf

$$\bar{F}_I(x) \equiv N^{-1} \sum_{i=1}^{N} I(I_i \leq x). \tag{5}$$

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

$$\bar{I}_N \equiv R^{-1} \sum_{i=1}^{R} N^{-1} \sum_{i=1}^{N} I_i, \tag{6}$$

where R is the number of replications.

3.3. 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

$$Z_{k+1} = Z_k + \rho N_k/\delta - (1+c_a^2)/\delta, \quad k \geq 1, \quad Z_0 \equiv 0. \tag{7}$$

Clearly, each step in (2) is divided by δ in (7). 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 (3) The number of customers served in that busy cycle, N_c, and the length of a busy cycle, C, are then

$$N_c = \inf\{k \geq 1 : Z_k \leq 0\} \quad \text{and} \quad C = N_c(1+c_a^2)\delta. \tag{8}$$

The associated idle-time random variable is thus distributed as

$$I \stackrel{\mathrm{d}}{=} -\delta Z_{N_c}. \tag{9}$$

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

$$0 \leq -Z_{N_c} \leq \frac{1+c_a^2-\rho}{\delta} \quad \text{and} \quad 0 \leq I \leq 1+c_a^2-\rho. \tag{10}$$

Given the alternative recursion in (7), 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 $\mathbb{E}[I^2]$.

3.4. Numerical Experiments for the DTMC Algorithm

To illustrate the DTMC numerical algorithm, we consider the example with $c_a^2 = 4$. First, Table 4 shows the results of the DTMC numerical algorithm for two values of $\rho = 0.5$ and $\rho = 0.8$. 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 4 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
1×10^2	14.913166	14.904170	3.470132	3.470132	3.470163
5×10^2	14.916936	14.916816	3.470132	3.470132	3.470132
1×10^3	14.916937	14.916936	3.470132	3.470132	3.470132
2×10^3	14.916937	14.916937	3.470132	3.470132	3.470132
5×10^3	14.916937	14.916937	3.470132	3.470132	3.470132

Table 4 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 5×10^3. 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 5 shows the performance of the DTMC algorithm as a function of N for other ρ. The appropriate δ is used in each case.

Table 5 Performance of DTMC Algorithm for Other Traffic Levels

$N\backslash\rho$	0.95	0.90	0.70	0.60	0.40	0.30
1	74.512312	34.621172	8.372901	5.243412	2.289971	1.493015
10	74.512312	34.696376	8.381077	5.267151	2.296621	1.498390
1×10^2	74.568945	34.719782	8.434009	5.294671	2.304104	1.499233
5×10^2	74.608460	34.719782	8.441300	5.294825	2.304105	1.499234
1×10^3	74.616306	34.721369	8.441305	5.294825	2.304105	1.499234
2×10^3	74.619898	34.721484	8.441305	5.294825	2.304105	1.499234
5×10^3	74.620917	34.721484	8.441305	5.294825	2.304105	1.499234
1×10^4	74.620917	34.721484	8.441305	5.294825	2.304105	1.499234

Finally, Table 6 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 (10). We report the results for different N.

Table 6 Performance of DTMC(N) for $\rho = 0.99$

$\delta\backslash N$	1×10^2	5×10^2	1×10^3	2×10^3	3×10^3
0.01	394.420259	394.476457	394.496173	394.511729	394.518208
$\delta\backslash N$	5×10^3	1×10^4	2×10^4	4×10^4	6×10^4
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 10^5 seconds CPU time for $N \geq 2\times10^4$ to attain six decimal places accuracy for $\rho = 0.99$. In contrast, with only 7×10^3 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 $N = 2\times10^3$.

4. More about Simulation Algorithms

We now describe the simulation algorithms in more detail.

4.1. 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 equation (1) of the main paper. 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

$$\bar{W} \equiv \bar{W}(N) \equiv N^{-1} \sum_{n=1}^{N} W_n. \tag{11}$$

From equation (2) of the main paper, 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 Asmussen and Glynn (2007). 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.

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 Asmussen and Glynn (2007) and §XIII.7 of Asmussen (2003). Moreover, the standard simulation method is not efficient under heavy traffic levels because of its slow convergence; e.g., see Whitt (1989).

4.2. Simulation 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

$$\bar{W}_R \equiv R^{-1} \sum_{i=1}^{R} \bar{W}_{[i]}, \tag{12}$$

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

$$S^2 \equiv (1/(R-1)) \sum_{i=1}^{R} (\bar{W}_{[i]} - \bar{W}_R)^2, \tag{13}$$

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.

4.3. Simulation Efficiency.

We compared the simulation efficiency of the three simulation algorithms in Table 6 of the main paper. We now elaborate by providing additional simulation results. 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 (Typically, in the Minh and Sorli (1983) simulation algorithm, we implement discrete event simulation. 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.). 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 7.

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 8. Table 8 shows the large error for smaller N with MC, but no problem at all with RW.

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 10^{-4} confidence interval length for most of the traffic levels, while the MC can only have 10^{-3} confidence interval length.

Expressed differently, in order to achieve 10^{-3} or 10^{-2} 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 $N = 10^6$ 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.

Table 7 A Comparison of Three Simulation Methods

Confidence Interval Length for the MC method as a Function of N, R and ρ

	$N=5\times10^4$			$N=1\times10^5$			$N=1\times10^6$		
$R\backslash\rho$	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

	$N=1\times10^2$			$N=5\times10^2$			$N=1\times10^3$		
$R\backslash\rho$	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

	$T=1\times10^3$			$T=1\times10^4$			$T=1\times10^5$		
$R\backslash\rho$	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

Table 8 A Comparison between MC and RW Simulation for $\rho = 0.99$

	$N = 1 \times 10^2$	$N = 1 \times 10^2$	$N = 5 \times 10^2$	$N = 5 \times 10^2$	$N = 1 \times 10^3$	$N = 1 \times 10^3$
$R = 100$	$\mathbb{E}[W]$	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
	$N = 5 \times 10^4$	$N = 5 \times 10^4$	$N = 1 \times 10^5$	$N = 1 \times 10^5$	$N = 1 \times 10^6$	$N = 1 \times 10^6$
$R = 100$	$\mathbb{E}[W]$	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

5. Additional Simulation Experiments

In order to better understand the computational 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.

5.1. 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 9 A Comparison of Monte-Carlo simulation for Two Queues

Confidence Interval Length for MC for $F_0/G_u/1$ with $M_s = 10^3$									
	$N = 5E+04$			$N = 1E+05$			$N = 1E+06$		
$R\backslash\rho$	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 9 shows that the model reduction makes the Monte-Carlo simulation more efficient and accurate. Typically, the simulation is most accurate for $F_0/D/1$.

5.2. 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 §3 of the main paper. So we compare the simulation performance for $F_0/G_u/1$ with given $M_s = 10^3$, $F_0/D/1$ and the queue $D/RS(\rho, p)/1$.

Table 10 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 9 and 10 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.

5.3. The Idle-Time Distribution in Two Queues.

We apply the Minh and Sorli (1983) 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 11.

Table 11 **A Comparison of the idle-time Distribution in the $F_0/G_{u^*}/1$ and $M/M/1$ queues, using the Minh and Sorli (1983) 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

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 $F_0/D/1$.

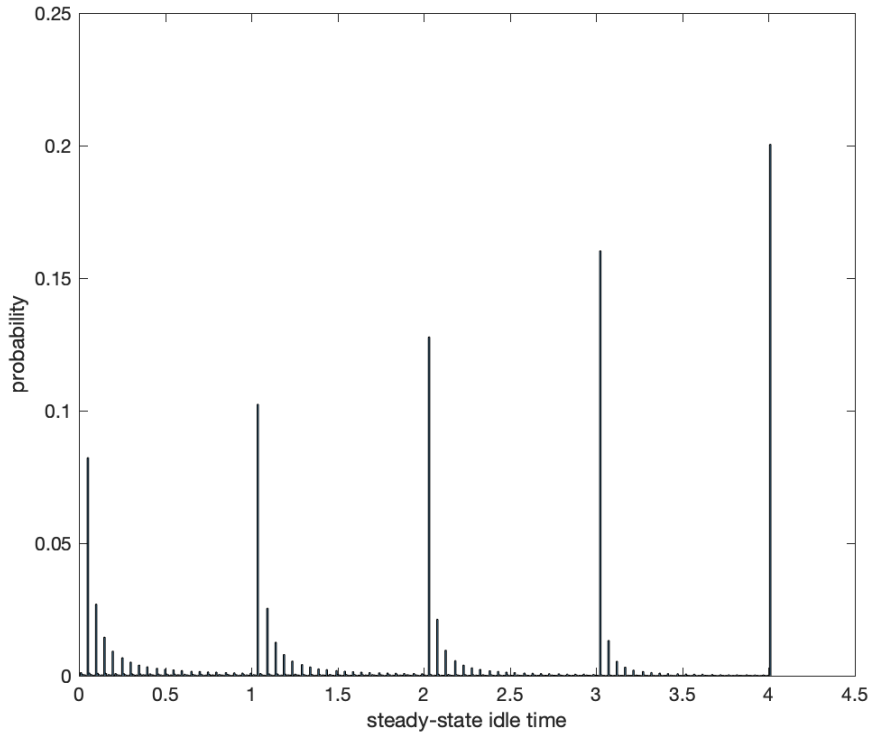

Figure 1 **Simulation estimates of the steady-state idle-time distribution in the $F_0/G_{u^*}/1$ model under traffic level $\rho = 0.99$.**

References

Asmussen S (2003) *Applied Probability and Queues* (New York: Springer), second edition.

Asmussen S, Glynn PW (2007) *Stochastic Simulation: Algorithms and Analysis* (New York: Springer), second edition.

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

Chen Y, Whitt W (2019) Algorithms for the upper bound mean waiting time in the $GI/GI/1$ queue, Columbia University, http://www.columbia.edu/∼ww2040/allpapers.html.

Daley DJ (1977) Inequalities for moments of tails of random variables, with queueing applications. *Zeitschrift fur Wahrscheinlichkeitsetheorie Verw. Gebiete* 41:139–143.

Kingman JFC (1962) Inequalities for the queue $GI/G/1$. *Biometrika* 49(3/4):315–324.

Minh DL, Sorli RM (1983) Simulating the $GI/G/1$ queue in heavy traffic. *Operations Research* 31(5):966–971.

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