EXACT SAMPLING FOR HETEROGENEOUS MULTISERVER JOB MODEL

Alexander S. Golovin

•

Institute of Applied Mathematical Research, KarRC RAS, Russia,
Petrozavodsk State University, Russia,
golovin@krc.karelia.ru

Abstract

The paper presents an approach to the simulation of a multi-server queueing system, known as the multi-server job model, where the (random number of) servers are seized/released by a customer simultaneously. This model is widely used in scenarios like cloud computing and parallel processing. Due to the inherent difficulty in obtaining analytical solutions for such systems, we adopt the exact sampling technique to generate the steady-state workload samples accurately. This technique allows one to obtain unbiased estimates of the steady-state performance, such as the per-class waiting times of customers.

Keywords: Exact sampling, dominated coupling from the past, multiserver model with simultaneous service, modified Kiefer-Wolfowitz workload vector.

1. Introduction

Since 1965, the multiserver job model (MJM) has been investigated, in which a customer may need simultaneous service on one or more servers [1]. Such models naturally arise in many applications including telecommunication, computer and emergency systems. In particular, a communication system with multiple-address messages, where a message is transmitted simultaneously to many receivers, was treated in [2]. Computer systems with competition for storage space and programs having diverse storage size and time requirements were considered in the paper [3] using a loss MJM, where jobs which cannot immediately receive service are dropped. Analytical results were provided for M/M/2-type model in papers [4, 5]. In [6], using a matrix-analytical approach, a criterion was obtained for the stability of M/M/c-type model. Much of the recent research on MJM is motivated by the need to understand and optimize the performance of data centers and cloud computing infrastructure, see, e.g. [7, 8, 9]. The research directions in MJM in recent years are described in sufficient detail in papers [10, 11]. While extensive literature exists on the MJM, the general case characterized by an arbitrary service time distribution and more than two servers remains largely unexplored, necessitating further investigation.

When dealing with complex queueing systems, often with non-standard arrival or service processes, analytical derivations of the performance characteristics can become intractable or impossible. *Exact sampling* (perfect simulation) techniques [12] provide powerful simulation-based alternative. Those allow one to obtain samples directly from the steady-state distribution, avoiding the need to wait for sufficiently long "warm-up" period during simulation.

This paper is dedicated to adopting the Dominated Coupling From the Past (DCFTP) exact sampling technique to the MJM. In [13] exact sampling technique was applied to MJM, to the best of our knowledge, for the first time, where the cases considered included heavy-tailed (Pareto), phase-type and exponential service time distributions. However, the regenerative exact simulation approach suggested in [13] has serious technical issues with (on average) infinite number of iterations while there is no such drawback in the approach proposed in the present paper.

We develop our approach on the basis of [14]. In the paper [14] the DCFTP algorithm [15] was adopted to sampling from the stationary customer delay in First-Come-First-Served (FCFS) M/G/c queue in the "super-stable" case, when the system load $\rho < 1$ (we call this scheme *the original algorithm* below, for short). Note that this condition on the load is heavily restrictive for the classical multiserver queue, since in general the stability requires $\rho < c$ only. At the same time, as shown in the present paper, the condition $\rho < 1$ is reasonable for the MJM case, which advocates the applicability of the proposed approach. So, our key contribution is a novel adaptation of the DCFTP algorithm to *heterogeneous* MJM which, in contrast to [13], guarantees finite average runtime.

The structure of the paper is as follows. In Section 2, we discuss the original approach to exact simulation [14] of the stationary customer delay in the M/G/c-FCFS queue based on the general DCFTP method where the single-server M/G/1 queue under the processor sharing (PS) discipline serves as an upper bound. Description of the MJM and modifications (for class-independent and class-dependent service times) of the original algorithm are given in Section 3. In Section 4, we present the results of numerical experiments and validation, which, in particular, include the comparison with an exact result obtained in [4] for the M/M/2-type model. We end up with a conclusion discussing possible further research directions.

2. Original Algorithm

This section is devoted to a detailed description of the *original algorithm* introduced in [14].

Consider a stable M/G/c-FCFS system Σ with arrival rate λ, Poisson arrival times $\{t_n : n \geq 0\}$ ($t_0 := 0$). Let the independent and identically distributed (iid) service times $\{S_n\}$ have distribution function (d.f.) $G(x) = P(S \leq x)$, $x \geq 0$, and finite mean $\mathrm{E}S = 1/\mu$. The workload vector $\boldsymbol{V}(t) = (V^{(1)}(t), V^{(2)}(t), \dots, V^{(c)}(t))$, $t \geq 0$, contains the residual work at the corresponding servers at time t, in ascending order, $V^{(1)}(t) \leq V^{(2)}(t) \leq \cdots \leq V^{(c)}(t)$. The value $\boldsymbol{V}(t)$ just before the arrival epoch is known as the Kiefer-Wolfowitz vector, $\boldsymbol{W}_n = \boldsymbol{V}(t_n-)$, and satisfies the recursion

$$\boldsymbol{W}_{n+1} = R\big(\boldsymbol{W}_n + S_n\boldsymbol{e} - T_n\boldsymbol{1}\big)^+ \tag{1}$$

where $T_n = t_{n+1} - t_n$ are iid interarrival times, $\boldsymbol{W}_n = (W_n^{(1)}, \dots, W_n^{(c)})$, $\boldsymbol{e} = (1, 0, \dots, 0)$, $\boldsymbol{1} = (1, 1, \dots, 1)$, R places the coordinates in ascending order, and $(\cdot)^+$ denotes the positive part of each coordinate. If $\rho := \lambda/\mu < c$, with $n \to \infty$, the vector $\boldsymbol{W}_n$ converges in distribution to the stationary workload vector $\boldsymbol{W}$ having distribution which coincides with the time-stationary limiting distribution of $\boldsymbol{V}(t)$ as $t \to \infty$ by the property PASTA.

The original algorithm for sampling exactly from the stationary workload vector $\boldsymbol{W}$ is proposed in [14] for the case

$$\rho < 1. \tag{2}$$

In this case, the system Σ is called super-stable. The algorithm is based on the following stochastic comparison result between the workload in the system Σ and the system M/G/1 operating under the processor sharing (PS) discipline (we denote it by $\hat{\Sigma}$). It is known (see e.g. [14]) that the workload in M/G/1-FCFS is an upper bound for the summary workload in Σ. Moreover, the workload in the system M/G/1 is invariant under changes of work-conserving disciplines [16], that is, workload has exactly the same sample paths under PS as it does under FCFS. Thus, the $\hat{\Sigma}$ serves as the upper bound of Σ [14]. The choice of PS discipline in the single-server queue is motivated by the fact that in this case the workload process turns out to be time reversible [17]. Note that under condition (2), the system $\hat{\Sigma}$ is stable. To apply the algorithm, it is necessary to have the steady-state distribution explicitly available, or we should have the possibility to draw from the statistical equilibrium. To ensure the finite completion time of the algorithm, we assume that $\mathrm{E}S^2 < \infty$ [14] (hereafter, unless stated otherwise, we omit the index for a typical member of the iid sequence).

The algorithm can be briefly described in the following steps:

1. From the stationary state, simulate the system $\hat{\Sigma}$ backward in time until it empties.
2. Invert the time axis and using the data of customer departure epochs (as arrivals) and corresponding work amounts from the trajectory of $\hat{\Sigma}$, construct the workload vectors in Σ by recursion (1).

The original approach is summarized in Algorithm 1. To simplify its comprehension, let's clarify some points. In the first step of the algorithm, we must start the workload process in $\hat{\Sigma}$ from the steady state. The stationary workload for the M/G/1 queue can be constructed from the Pollaczek-Khintchine formula as follows,

$$\sum_{i=1}^{Q} S_e^{(i)}, \tag{3}$$

where the $\left\{S_e^{(i)}\right\}_{i\geq 1}$ are iid residual *work amounts* having distribution function given by

$$G_e(x) = \frac{1}{\mathrm{E}S}\int_0^x \mathrm{P}(S > y)dy, \quad x \geq 0, \tag{4}$$

and, independently, Q has a geometric distribution

$$P(Q = k) = \rho^k(1-\rho), \quad k \geq 0. \tag{5}$$

However, in order to move backwards in time, and to properly restore the workload vector for the Σ by recursion (1) forward in time, we also need the complete (the sum of completed and residual) work amounts for the Q customers present in the system $\hat{\Sigma}$ at origin. Each such complete work amount H has the stationary spread distribution [18] with tail d.f. given by

$$P(H > x) = \frac{1}{\mathrm{E}S} x\overline{G}(x) + \overline{G}_e(x). \tag{6}$$

Note that in the system $\hat{\Sigma}$ we must distinguish the work amount of customer (which is defined upon arrival) and the service time (which indeed depends on the number of customers in the system during service). Indeed, in the $\hat{\Sigma}$ the customer starts being served immediately upon arrival, at a rate inversely proportional to the number of customers in the system. in Σ, on the contrast, service times and work amounts coincide.

Using a complete work amount H distributed as (6), we can generate a residual work amount S_e (having d.f. (4)) by the following trick [14]: take U uniform in $[0,1]$ and take $S_e = UH$. Using this trick, we generate Q iid copies H_i and iid U_i, and set $S_e^{(i)} = U_i H_i$, $i = 1, \ldots, Q$ to initialize the steady-state workload vector in $\hat{\Sigma}$. We also store the values $\{H_i\}_{i=1,\ldots,Q}$ used for the workload reconstruction in Σ at the second step.

After initialization, we simulate the workload in $\hat{\Sigma}$, say, by the discrete-event simulation. In fact, this simulation is done in reverse time, so we need to keep track of the customer departure epochs $-t_i$ and the corresponding work amounts S_i. Note that, after time inversion the customer departure epochs $-t_i$ will become arrival epochs and work amounts S_i have the d.f. G except the Q customers present in the system at origin, if any. We stop simulation at the time epoch $-t_n$ with $n = \min\{k \geq 0 : Q(-t_k) = 0\}$, where $Q(t)$ is the number of customers in $\hat{\Sigma}$ at time t. Technically this simulation can be implemented in "forward" time and then time reversal can be performed (care must be taken regarding the indices).

We also note that the service speed in system $\hat{\Sigma}$ changes both at departure and arrival epochs. Actually, the server processes work at unit rate, which is divided equally among all customers present in the system. That is, whenever there are Q customers in the system, each will receive service at rate $r = 1/Q$.

At the second step, we construct the Kiefer-Wolfowitz vector for Σ. We need to correctly define the interarrival times in Σ and the initial conditions for the workload vector $\boldsymbol{W}_1$. To do so, we firstly define the interdeparture times (in reverse time) in $\hat{\Sigma}$ as $T_i = t_i - t_{i-1}$, $1 \le i \le n$, where $t_0 = 0$, and write down the corresponding work amounts $S_1, \dots, S_n$. Then time reversal of $\hat{\Sigma}$ is performed by resetting $(T_1, \dots, T_n) := (T_n, \dots, T_1)$ and $(S_1, \dots, S_n) := (S_n, \dots, S_1)$, respectively. Thus, sequences $(T_1, \dots, T_n)$ and $(S_1, \dots, S_n)$ are now the sample path in $\hat{\Sigma}$ from an empty state at time in the past $-t_n$, $t_n \ge 0$. Starting from $\boldsymbol{W}_1 = \mathbf{0}$, we reconstruct the sample path in Σ up to $\boldsymbol{W}_{n+1}$ which is taken as the sample from the steady state. Indeed, if at time $-t_n$ the first customer arrives in an empty system and until time 0 we have n arrivals, then at time 0 the system will be observed by a customer with the number $n+1$.

Algorithm 1 The original approach

```
Require: Service time d.f. $G$, arrival rate $\lambda$
Set modeling time $t = t_0 = 0$
Set $n = 0$
Generate the number of customers, $Q$, in the system $\hat{\Sigma}$ using (5).
if $Q = 0$ then
    Set $\boldsymbol{V}(0) = \mathbf{0}$
else
    Sample iid $H_i$, $i = 1, \dots, Q$ from the stationary spread distribution, using (6)
    Sample iid $U_i$, $i = 1, \dots, Q$, uniformly on $[0, 1]$
    Take $\{Q, U_1 H_1, \dots, U_Q H_Q\}$ as the stationary workload in $\hat{\Sigma}$ system at origin
    Sample interarrival time $T \sim Exp(\lambda)$
    while $Q > 0$ do                                  ▷ Discrete-event simulation of $\hat{\Sigma}$
        if the next event is an arrival then
            Generate a work amount $S \sim G$ and keep a record of the value
            Sample interarrival time $T \sim Exp(\lambda)$
            Reset $Q = Q + 1$ and set $r = 1/Q$
        else if the next event is a departure then
            Reset $n = n + 1$
            Record the departure time $t_n$ of the customer
            Reset $Q = Q - 1$ and set $r = 1/Q$
        end if
        Update the times and residual work amounts
    end while
    Define the interdeparture times $T_i = t_i - t_{i-1}$, $1 \le i \le n$
    Reset $(S_1, \dots, S_n) = (S_n, \dots, S_1)$ and $(T_1, \dots, T_n) = (T_n, \dots, T_1)$
    Set $\boldsymbol{W}_1 = \mathbf{0}$
    for $i = 1, 2, \dots, n$ do                       ▷ Reconstruct the sample path in $\Sigma$
        $\boldsymbol{W}_{i+1} = R\big(\boldsymbol{W}_i + S_i \boldsymbol{e} - T_i \mathbf{1}\big)^+$
    end for
    Set $\boldsymbol{V}(0) = \boldsymbol{W}_{n+1}$
end if
return $\boldsymbol{V}(0)$
```

3. Modified Algorithm for Heterogeneous MJM

Let us describe a MJM with c servers operating under FCFS discipline [4, 19, 20, 6]. In such a system, class-k customer requires k (available) servers which are seized and released simultaneously, $1 \le k \le c$. The (random) service time S_i of customer i is thus identical at each of the N_i (random number of) servers. Note that we consider a heterogeneous MJM system (denoted by Σ) where $\{S_i\}_{i \ge 1}$ may have a class-dependent distribution.

The input of class-k customers follows a Poisson process with rate λ_k. Thus, the interarrival times $\{T_i\}_{i\geq 1}$ have an exponential distribution with rate $\lambda = \lambda_1 + \cdots + \lambda_c$, and

$$p_k = \mathrm{P}\{N = k\} = \lambda_k/\lambda \tag{7}$$

may be treated as the probability of class-k customer arrival.

In MJM, servers can be idle when the head-of-queue customer blocks the subsequent customers in the queue, which is known as a non-work-conserving property. After appropriate modification of the Kiefer–Wolfowitz recursion (1), the workload vector can be calculated as follows [20]:

$$\boldsymbol{W}_{i+1} = R\big((W_{i,N_i} + S_i)\boldsymbol{e}_{1:N_i} + \boldsymbol{W}_i \circ (\mathbf{1} - \boldsymbol{e}_{1:N_i}) - T_i\mathbf{1}\big)^+, \tag{8}$$

where $\boldsymbol{e}_{1:k}$, for $k = 1, \ldots, c$, is a vector having ones on the first k elements and zeroes elsewhere and $\circ$ is Hadamard (componentwise) multiplication. Alternatively, (8) reads

$$\boldsymbol{W}_{i+1} = R\big(\underbrace{W_{i,N_i} + S_i - T_i, \ldots, W_{i,N_i} + S_i - T_i}_{N_i \text{ components}}, W_{i,N_i+1} - T_i, \ldots, W_{i,c} - T_i\big)^+.$$

We can see that the first N_i components of the vector are the same, since customer i waits for the release of exactly N_i servers and seizes them simultaneously for the same time S_i. Thus W_{i,N_i} is the delay of the customer i, and W_N is the (stationary) delay of a generic customer of class N. By this reason, each coordinate in the stationary workload vector $\boldsymbol{W}$ gives the stationary delay of the customer of the corresponding class.

To appropriately modify the original algorithm for the MJM, we need to find a suitable upper bound $\hat{\Sigma}$. Firstly, we need the following intermediate result. It was shown in [20] that for the MJM in homogeneous case (class-independent service times), an upper bound $\tilde{\Sigma}$ can be constructed as follows. By coupling, take the same interarrival times $\tilde{T}_i = T_i$ and service times $\tilde{S}_i = S_i$, $i \geq 1$. In the c-server system $\tilde{\Sigma}$, force each customer to use all the c servers simultaneously. In such a case, workload $\tilde{\boldsymbol{W}}_i$ at each of the c servers upon arrival of customer i essentially follows the Lindley recursion,

$$\tilde{\boldsymbol{W}}_{i+1} = \big(\tilde{\boldsymbol{W}}_i + S_i\mathbf{1} - T_i\mathbf{1}\big)^+. \tag{9}$$

It can be seen that this c-server system is in fact equivalent to an M/G/1 system operating under FCFS discipline. Moreover, the proof in [20] does not depend on the assumption of homogeneity for the service times. However, a more delicate coupling is needed in the heterogeneous case, which we provide below.

Take c iid sequences $\{S_i^{(k)}\}_{i\geq 1}$, where the generic element $S^{(k)}$ has d.f. $G^{(k)}$, $k = 1, \ldots, c$. The service times in Σ are taken in accordance with the class N_i of the customer i as $S_i^{(N_i)}$, i.e. the i-th element of the appropriate sequence. In this case, the upper bound for the workload sequence $\{W_i\}_{i\geq 1}$ is provided by the c-server system $\tilde{\Sigma}$ in which the workload sequence $\{\tilde{W}_i\}_{i\geq 1}$ follows the recursion (9), where the service time of the i-th customer is given as a *finite mixture*

$$\tilde{S}_i = \sum_{k=1}^{c} I_i^{(k)} S_i^{(k)}, \tag{10}$$

and $I_i^{(k)}$ is the indicator that the i-th customer has class k. Note that $I_i^{(N_i)} = 1$ implies $\tilde{S}_i = S_i^{(N_i)}$, so the service times of customers in systems Σ and $\hat{\Sigma}$ coincide. Thus the system $\tilde{\Sigma}$ can be used as an upper bound for the system Σ (in which we indeed use a coupling to have relations w.p.1 between random variables) under the conditions given in the following Lemma.

Lemma 1. Assume that in the systems Σ and $\tilde{\Sigma}$, the initial workload vectors are ordered as $\boldsymbol{W}_1 \leq \tilde{\boldsymbol{W}}_1$, and the governing variables are related as $T_i = \tilde{T}_i$, $\hat{S}_i = S_i^{(N_i)}$, $i = 1, 2, \ldots$. Then $\boldsymbol{W}_i \leq \tilde{\boldsymbol{W}}_i$.

Proof. Assume by induction that $\boldsymbol{W}_i \leq \tilde{\boldsymbol{W}}_i$ for some $i \geq 1$. Rewrite (8) as follows,

$$\boldsymbol{W}_{i+1} = \left\{ \begin{array}{c} W_{i+1,1} \\ \cdots \\ W_{i+1,c} \end{array} \right\} = R \left\{ \begin{array}{c} W_{i,N_i} + S_i^{(N_i)} - T_i \\ \cdots \\ W_{i,N_i} + S_i^{(N_i)} - T_i \\ W_{i,N_i+1} - T_i \\ \cdots \\ W_{i,c} - T_i \end{array} \right\}^+ .$$

Since the operator $R(\cdot)^+$ keeps the ordering between components (e.g. see Lemma 1 [20]) and using (9), verify

$$\boldsymbol{W}_{i+1} = \left\{ \begin{array}{c} W_{i+1,1} \\ \cdots \\ W_{i+1,c} \end{array} \right\} = R \left\{ \begin{array}{c} W_{i,N_i} + S_i^{(N_i)} - T_i \\ \cdots \\ W_{i,N_i} + S_i^{(N_i)} - T_i \\ W_{i,N_i+1} - T_i \\ \cdots \\ W_{i,c} - T_i \end{array} \right\}^+ \leq R \left\{ \begin{array}{c} W_{i,c} + S_i^{(N_i)} - T_i \\ \cdots \\ W_{i,c} + S_i^{(N_i)} - T_i \\ W_{i,c} - T_i \\ \cdots \\ W_{i,c} - T_i \end{array} \right\}^+ \leq$$

$$R \left\{ \begin{array}{c} W_{i,c} + S_i^{(N_i)} - T_i \\ \cdots \\ W_{i,c} + S_i^{(N_i)} - T_i \end{array} \right\}^+ \leq \left\{ \begin{array}{c} \tilde{W}_i + S_i^{(N_i)} - T_i \\ \cdots \\ \tilde{W}_i + S_i^{(N_i)} - T_i \end{array} \right\}^+ = \left\{ \begin{array}{c} \tilde{W}_{i+1} \\ \cdots \\ \tilde{W}_{i+1} \end{array} \right\}^+ = \tilde{\boldsymbol{W}}_{i+1}.$$

■

We have shown that $\tilde{\Sigma}$ is a componentwise upper bound for Σ in terms of the workload. At the same time, $\tilde{\Sigma}$ is equivalent to M/G/1-FCFS which is insensitive to the work-conserving queueing discipline. Thus, we can take the workload sequence $\{\hat{W}_i\}_{i\geq 1}$ in the M/G/1-PS system $\hat{\Sigma}$ as an upper bound for each component of the workload vector sequence $\{\boldsymbol{W}_i\}_{i\geq 1}$.

Now we need to construct the steady state of the system $\hat{\Sigma}$. Denote the class-k service rate by $\mu_k = 1/ES^{(k)}$ and the class-k load by $\rho_k = \lambda_k/\mu_k$, $k = 1, \ldots, c$. Let $H^{(k)}$ have the stationary spread distribution (6) for each class k (in obvious notation). A construction of the spread distribution in $\hat{\Sigma}$ is given in the following Lemma.

Lemma 2. Let H have the stationary spread distribution in $\hat{\Sigma}$, then

$$P(H > x) = \sum_{k=1}^{c} \frac{\rho_k}{\rho} P(H^{(k)} > x). \tag{11}$$

Proof. Indeed, it follows from (4), (6) and (10), after simple algebra,

$$P(H > x) = \mu x \overline{G}(x) + \overline{G}_e(x) = \mu x \textstyle\sum_{k=1}^{c} p_k \overline{G}^{(k)}(x) + \mu \sum_{k=1}^{c} \frac{p_k}{\mu_k} \overline{G}_e^{(k)}(x) =$$
$$= \textstyle\sum_{k=1}^{c} \frac{\rho_k}{\rho} \left(x\mu_k \overline{G}^{(k)}(x) + \overline{G}_e^{(k)}(x) \right) = \sum_{k=1}^{c} \frac{\rho_k}{\rho} P(H^{(k)} > x).$$

■

It follows from (11) that the spread distribution in $\hat{\Sigma}$ is a finite mixture of the corresponding class-k r.v. $H^{(k)}$, with mixing probabilities

$$\left\{ \frac{\rho_1}{\rho}, \ldots, \frac{\rho_c}{\rho} \right\}. \tag{12}$$

Thus, the stationary workload in $\hat{\Sigma}$ is constructed as

$$\left\{ Q, U_1 H_1^{(N_1)}, \ldots, U_Q H_Q^{(N_Q)} \right\}, \tag{13}$$

where Q has a geometric distribution (5), N_i are iid samples from (12), $H_i^{(N_i)}$ are iid samples from the corresponding stationary spread distributions, using (6) (in obvious notation), and U_i are iid samples from uniform distribution on $[0,1]$, $i = 1,\dots,Q$.

We are ready to describe the modified algorithm. At the first step, the steady state of the M/G/1-PS system $\hat{\Sigma}$ is constructed as (13). The system is simulated backwards in time until it empties, and the classes of customers are stored.

At the second step, invert the time axis, and using the data of customer departure epochs (as arrivals) and corresponding work amounts and customer classes from the trajectory of $\hat{\Sigma}$, construct the workload vectors in Σ by recursion (8).

The modified algorithm for class-dependent (heterogeneous) MJM is presented as a pseudocode in Algorithm 2.

The procedure is simplified in the homogeneous MJM where the service times of customers $\{S_i\}_{i\geq 1}$ are iid and have arbitrary class-independent distribution G. In this case the first step of the original algorithm in Section 2 remains unchanged. At the second step, the sample path in Σ is restored using the recursion (8), where classes of customers are *sampled* from discrete distribution (7). The homogeneous version of the modified algorithm is presented as a pseudocode in Algorithm 3.

4. Validation and Numerical Results

To validate the modified algorithm, we use the explicit results obtained by Brill and Green [4] in the M/M/2-type MJM. In this model, class-k customers arrive at rate λ_k, $k = 1,2$, and service times have exponential distribution with class-independent rate μ. A notable proximity of the exact solution given in Example 3 [4] to the sample obtained by Algorithm 3 is shown in Figure 1.

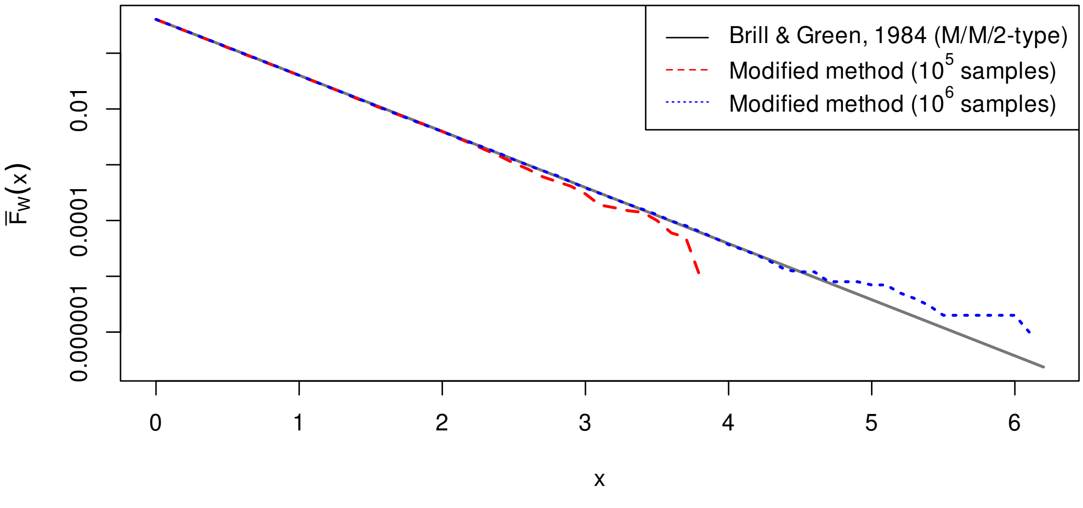

Figure 1: *M/M/2-type FCFS, $\lambda_1 = 1, \lambda_2 = 1, \mu = 4, \rho = 0.5$. Exact solution for tail d.f. waiting time (Example 3 [4]) and estimated c.d.f. from 10^5 and 10^6 samples obtained by Algorithm 3.*

In the following experiments we take the service time distribution as the second-type Pareto, ParetoII(α,β) having density and d.f.,respectively

$$f_S(x) = \frac{\alpha}{\beta}\left(\frac{\beta}{x+\beta}\right)^{\alpha+1}, \quad F_S(x) = 1 - \left(\frac{\beta}{x+\beta}\right)^{\alpha}. \tag{14}$$

Algorithm 2 The algorithm for heterogeneous MJM

```
Require: Service time d.f. $G^k$, arrival rates $\lambda_k$, $k = 1, \dots, c$
Set modeling time $t = t_0 = 0$
Set $n = 0$                                            ▷ order number of departure
Generate the number of customers, $Q$, in the system $\hat{\Sigma}$ using (5).
Set $m = Q$                                            ▷ order number of arrival
if $Q = 0$ then
    Set $\boldsymbol{V}(0) = \boldsymbol{0}$
else
    Sample iid $N_i$, $i = 1, \dots, m$ from discrete distribution $\{\rho_1/\rho, \dots, \rho_c/\rho\}$
    Sample iid $H_i^{(N_i)}$, $i = 1, \dots, m$ from the stationary spread distribution, using (6)
    Sample iid $U_i$, $i = 1, \dots, m$, uniformly on $[0, 1]$
    Take $\{Q, U_1 H_1^{(N_1)}, \dots, U_m H_m^{(N_m)}\}$ as the stationary workload in $\hat{\Sigma}$ system at origin
    Reset $m = m + 1$
    Sample customer class $N_m$ from discrete distribution $\{p_1, \dots, p_c\}$, record this value
    Sample a work amount $S_m \sim G^{(N_m)}$ and keep a record of the value
    Sample next interarrival time $T_m \sim Exp(\lambda_{N_m})$
    while $Q > 0$ do                                   ▷ Discrete-event simulation of $\hat{\Sigma}$
        if the next event is an arrival then
            Reset $m = m + 1$
            Sample class of arrival customer $N_m$ from discrete distribution $\{p_1, \dots, p_c\}$, record
    this value
            Sample a work amount $S_m \sim G^{(N_m)}$ and keep a record of the value
            Sample the next interarrival time $T_m \sim Exp(\lambda_{N_m})$
            Reset $Q = Q + 1$ and set $r = 1/Q$
        else if the next event is a departure then
            Reset $n = n + 1$
            Record the departure time $t_n$ of the customer
            Reset $Q = Q - 1$ and set $r = 1/Q$
        end if
    end while
    Define the interdeparture times $T_i = t_i - t_{i-1}, 1 \le i \le n$, with $t_0 = 0$
    Reset $(S_1, \dots, S_n) = (S_n, \dots, S_1)$, $(T_1, \dots, T_n) = (T_n, \dots, T_1)$ and $(N_1, \dots, N_n) =$
    $(N_n, \dots, N_1)$
    for $i = 1, 2, \dots, n$ do                        ▷ Reconstruct the sample path in $\Sigma$
        $\boldsymbol{W}_{i+1} = R\big((W_{i,N_i} + S_i)\boldsymbol{e}_{1:N_i} + \boldsymbol{W}_i \circ (\boldsymbol{1} - \boldsymbol{e}_{1:N_i}) - T_i\boldsymbol{1}\big)^+$
    end for
    Set $\boldsymbol{V}(0) = \boldsymbol{W}_{n+1}$
    Sample class of arrival customer $N_{n+1}$ from discrete distribution $\{p_1, \dots, p_c\}$
end if
return $(\boldsymbol{V}(0), N_{n+1})$
```

Algorithm 3 The algorithm for homogeneous MJM

```
Require: Service time d.f. $G$, arrival rates $\lambda_1, \dots, \lambda_c$
Set modeling time $t = t_0 = 0$
Set $n = 0$
Generate the number of customers, $Q$, in the system $\hat{\Sigma}$ using (5).
if $Q = 0$ then
    Set $\boldsymbol{V}(0) = \boldsymbol{0}$
else
    Sample iid $H_i$, $i = 1, \dots, Q$ from the stationary spread distribution, using (6)
    Sample iid $U_i$, $i = 1, \dots, Q$, uniformly on $[0, 1]$
    Take $\{Q, U_1 H_1, \dots, U_Q H_Q\}$ as the stationary workload in $\hat{\Sigma}$ system at origin
    Sample interarrival time $T \sim Exp(\lambda)$
    while $Q > 0$ do                                  ▷ Discrete-event simulation of $\hat{\Sigma}$
        if the next event is an arrival then
            Sample a work amount $S \sim G$ and keep a record of the value
            Sample interarrival time $T \sim Exp(\lambda)$
            Reset $Q = Q + 1$ and set $r = 1/Q$
        else if the next event is a departure then
            Reset $n = n + 1$
            Record the departure time $t_n$ of the customer
            Reset $Q = Q - 1$ and set $r = 1/Q$
        end if
        Update the times and residual work amounts
    end while
    Define the interdeparture times $T_i = t_i - t_{i-1}, 1 \le i \le n$
    Reset $(S_1, \dots, S_n) = (S_n, \dots, S_1)$ and $(T_1, \dots, T_n) = (T_n, \dots, T_1)$
    Sample classes of arrival customers $(N_1, \dots, N_{n+1})$ in $\Sigma$ from discrete distribution
      $\{p_1, \dots, p_c\}$
    Set $\boldsymbol{W}_1 = 0$
    for $i = 1, 2, \dots, n$ do                        ▷ Reconstruct the sample path in $\Sigma$
        $\boldsymbol{W}_{i+1} = R\big((W_{i,N_i} + S_i)\boldsymbol{e}_{1:N_i} + \boldsymbol{W}_i \circ (\boldsymbol{1} - \boldsymbol{e}_{1:N_i}) - T_i \boldsymbol{1}\big)^+$
    end for
    Set $\boldsymbol{V}(0) = \boldsymbol{W}_{n+1}$
end if
return $(\boldsymbol{V}(0), N_{n+1})$
```

The stationary waiting time distribution in M/ParetoII(α,β)/1 is obtained in [21] in explicit form,

$$F_W(x) = 1 - \rho(1-\rho)\int_0^\infty \frac{u^{\alpha-2}e^{-(1+x/\beta)u}}{\Gamma(\alpha-1)H_1(u,\alpha,\rho)}du, \tag{15}$$

where

$$H_1(u,\alpha,\rho) = \begin{cases} (1+(\alpha-1)\rho e^{-u}Ei_\alpha(u))^2 + (\rho I(u,\alpha-1))^2 \text{ if } \alpha = 2,3,... \\ (1-\rho R(u,\alpha-1))^2 + (\rho I(u,\alpha-1))^2 \text{ if } \alpha > 1 \cup \alpha \neq 2,3,..., \end{cases}$$

$Ei_\alpha(u)$ is Ramsay's generalization of the exponential integral,

$$I(u,\alpha) = \frac{\pi u^\alpha e^{-u}}{\Gamma(\alpha)}, \quad R(u,\alpha) = {}_1F_1(1,1-\alpha,-u) - I(u,\alpha)\cot(\pi\alpha),$$

and ${}_1F_1(1,1-\alpha,-u)$ is the hypergeometric function of the first kind.

In the first experiment, we apply the modified method to illustrate the stochastic monotonicity established in Lemma 1. We plot the tails of the per-class (as well as generic) waiting time distributions in a heterogeneous 2-server MJM. We take $\rho = 0.4$, $\lambda_1 = 0.832$, $\lambda_2 = 0.208$, and use class-dependent ParetoII(α_k,β_k) service time distribution for class-k customer, $k = 1,2$, where $\alpha_1 = 3.5$, $\alpha_2 = 4.1$, $\beta_1 = \beta_2 = 1$. Figure 2 shows that the tail of the waiting time distribution of a generic customer in a heterogeneous MJM 2-server system is sandwiched between the waiting time distributions of class 1 and class 2 customers. It is also shown that the classical M/G/2-FCFS system with a finite mixture service time distribution is the lower bound, and M/G/1-FCFS is the upper bound for the 2-server MJM. The lower bound d.f. estimate is built from samples obtained by the original method, and the upper bound obtained explicitly using (15).

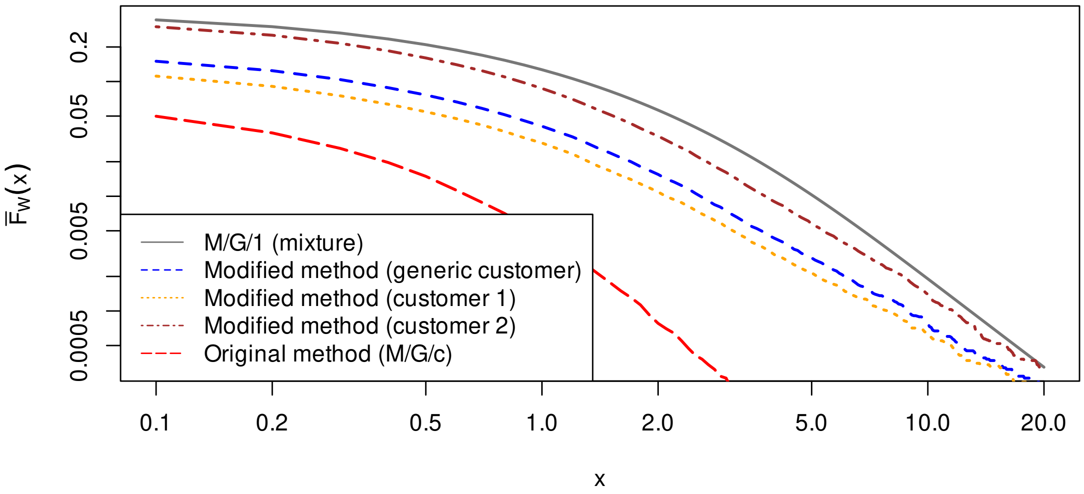

Figure 2: *Waiting time tail distribution for the case $c = 2, \rho = 0.4, \lambda_1 = 0.832, \lambda_2 = 0.208, \alpha_1 = 3.5, \alpha_2 = 4.1, \beta_1 = \beta_2 = 1, p_1 = 0.8, p_2 = 0.2$, where service time has distribution Pareto II. Explicit solution for M/G/1 for mixture vs. samples by modified method (over 10^5 samples) vs. original method (over 10^5 samples). Modified method (generic customer) is the tail c.d.f. delay which see a generic customer, modified method (customer 1) - customer of class 1, modified method (customer 2) - customer of class 2. Note the logarithmic scale on both axes.*

In the subsequent experiments we use the homogeneous MJM, when the service times are class-independent.

Figure 3 shows the results of an experiment with an MJM system having two servers, with $\rho = 0.9$, input rate $\lambda = 2.25$, and ParetoII(α,β) service time distribution with $\alpha = 3.5$, $\beta =$

1. The results of sampling from the stationary delay in a classical M/G/2 system using the original method (Algorithm 1) and delay in MJM 2-server system using the modified method (Algorithm 3), as well as the sample path-based estimates for the delays (using corresponding stochastic recursions (1) and (8)), are shown. The workload distribution for the upper bound M/G/1-FCFS system is obtained explicitly using (15). In our example, single-server and two-server customers are equally likely to arrive, $p_1 = p_2 = 0.5$. Interestingly, despite the fact that only about half of the customers require simultaneous service on two servers, the waiting time for a generic customer is close to the waiting time in a single-server system.

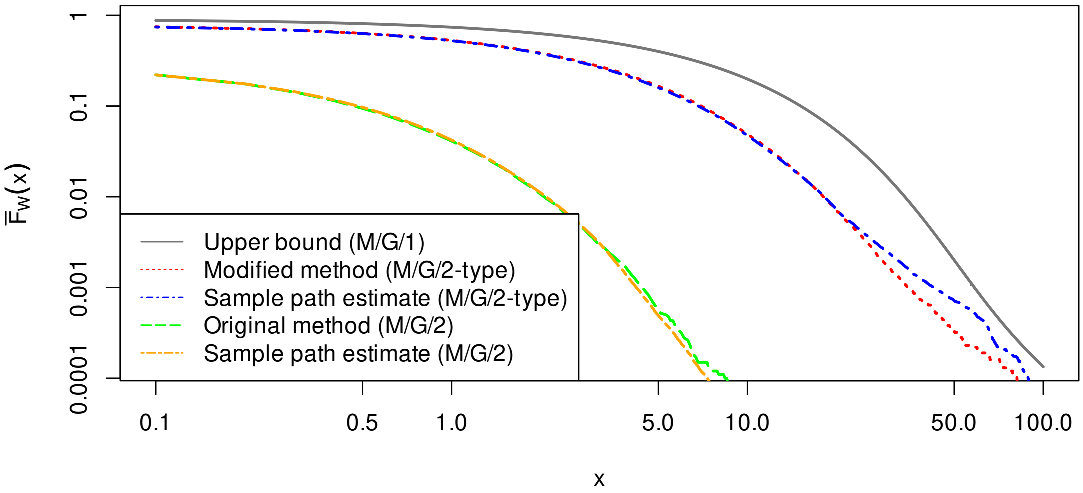

Figure 3: *Waiting time tail distribution for the case $c = 2$, $\rho = 0.9$, $\lambda = 2.25$, $\alpha = 3.5$, $\beta = 1$, where service time has distribution* (14)*. Explicit solution* (15) *for M/G/1 vs. samples by modified method* (10^5 *samples) vs. sample path of MJM* (10^6 *arrivals) vs. original method for classic M/G/2 model* (10^5 *samples) vs. sample path estimates for classic M/G/2 system* (10^6 *arrivals). Note the logarithmic scale on both axes.*

In Figure 4, we depict the estimated tail d.f. of the stationary delay (of a generic customer) in an MJM model with $c = 100$ servers, $\lambda = 2.25$, and ParetoII$(3.5, 1)$ service time distribution where all classes are sampled uniformly. A notable proximity of the d.f. with the explicit d.f. for M/G/1 model (15) is observed, which is in sharp contrast to the classical multiserver model. In fact, we performed several experiments with M/G/$\tilde{c}$ classical multiserver models, varying $\tilde{c}$ up to 10, and starting from $\tilde{c} = 7$ onwards, the delays are almost negligible. This illustrates that the condition $\rho < 1$ is not very restrictive for MJM (in contrast to the super-stability in the classical system).

It can be seen from Figures 3 and 4 that the sample path estimates give rather inaccurate results in the tail part. To study the reason for this inaccuracy, we compare the tail d.f. of the delays using an explicit solution (15), sample path estimate (10^6 customers), and simulation of stationary delay using formula (3) (10^5 samples) in M/G/1 system. The results depicted in Figure 5 show the same pattern which demonstrates that the number of samples for the sample path estimates should be taken large enough to mitigate the dependence of the members of a delay sequence.

5. Conclusion

In this paper we introduced a modification of the exact simulation approach suggested in [14]. Adaptation of this approach to heterogeneous MJM by using stochastic recursion in the form (8) allowed us to study the performance of this rather sophisticated model. The proposed algorithm allows one to obtain the unbiased estimates of the system performance, as well as the per-class

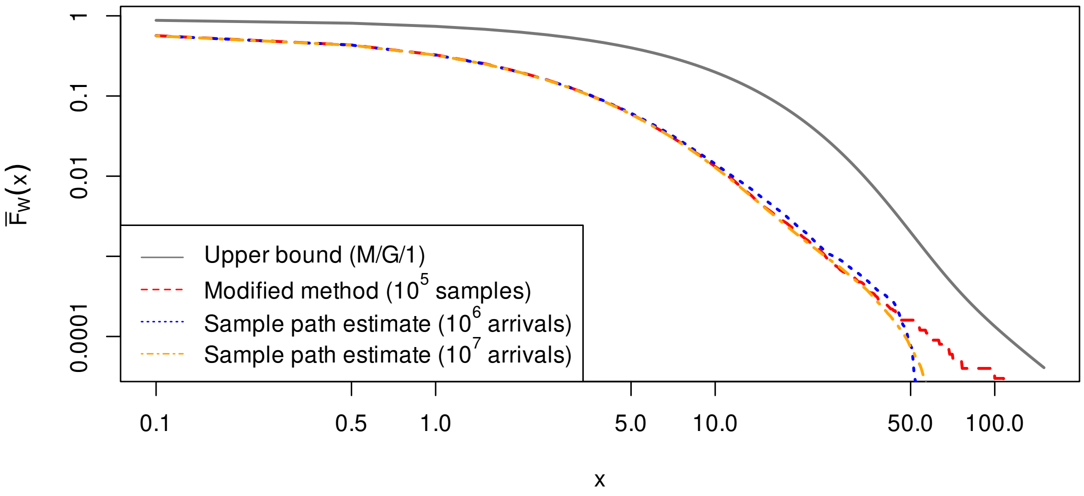

Figure 4: *Waiting time tail distribution for the M/G/100-type, where service time has distribution Pareto II, $\rho = 0.9$, $\lambda = 2.25$, $\alpha = 3.5$, $\beta = 1$. Explicit solution* (15) *for M/G/1 vs. samples by modified method* (10^5 *samples) vs. sample path estimates* (10^6 *and* 10^7 *arrivals). Note the logarithmic scale on both axes.*

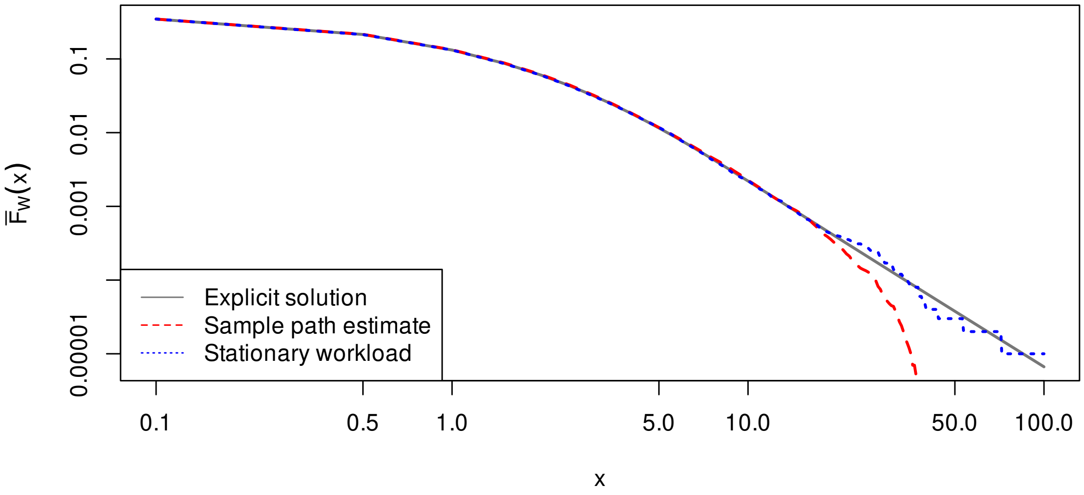

Figure 5: *Tail of the waiting time for M/G/1 model, where service time has distribution ParetoII(α, β), $\rho = 0.4$, $\lambda = 1$, $\alpha = 3.5$, $\beta = 1$. Explicit solution* (15) *vs. sample path estimate* (10^6 *arrivals) vs. simulation by* (3) (10^5 *samples). Note the logarithmic scale on both axes.*

stationary waiting times of customers. Heavy-tailed (type-II Pareto) distribution was used in the numerical experiments. As a further study, other service time distributions, such as exponential or phase-type distributions, can be considered, as well as another distribution for the customer class, such as the Zipf distribution, which may have practical value for other applications. In some situations, we are very interested in examining rare but critical events, such as the probability of very long waiting times. We suppose that exact sampling allows us to simulate these rare events, and, through a large number of samples, to accurately estimate the probability that such events will occur.

References

[1] L. Gimpelson. "Analysis of Mixtures of Wide- and Narrow-Band Traffic". en. In: *IEEE Transactions on Communications* 13.3 (1965), pp. 258–266. DOI: 10.1109/TCOM.1965.1089121.

[2] Eric Wolman. "The Camp-On Problem for Multiple-Address Traffic". en. In: *Bell System Technical Journal* 51.6 (1972), pp. 1363–1422. DOI: 10.1002/j.1538-7305.1972.tb02657.x.

[3] E. Arthurs and J. S. Kaufman. "Sizing a Message Store Subject to Blocking Criteria". In: *Proceedings of the Third International Symposium on Modelling and Performance Evaluation of Computer Systems: Performance of Computer Systems*. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1979, pp. 547–564. URL: http://dl.acm.org/citation.cfm?id=647409.724583.

[4] Percy H. Brill and Linda Green. "Queues in which customers receive simultaneous service from a random number of servers: a system point approach". In: *Management Science* 30.1 (1984), pp. 51–68. DOI: 10.1287/mnsc.30.1.51.

[5] D. Filippopoulos and H. Karatza. "An M/M/2 parallel system model with pure space sharing among rigid jobs". In: *Mathematical and Computer Modelling* 45.5–6 (2007), pp. 491–530. DOI: 10.1016/j.mcm.2006.06.007.

[6] Alexander Rumyantsev and Evsey Morozov. "Stability criterion of a multiserver model with simultaneous service". In: *Annals of Operations Research* 252.1 (2017), pp. 29–39. DOI: 10.1007/s10479-015-1917-2.

[7] Mor Harchol-Balter. "The multiserver job queueing model". en. In: *Queueing Systems* 100.3-4 (2022), pp. 201–203. DOI: 10.1007/s11134-022-09762-x.

[8] Diletta Olliaro et al. "The Impact of Service Demand Variability on Data Center Performance". en. In: *IEEE Transactions on Parallel and Distributed Systems* 36.2 (2025), pp. 120–132. DOI: 10.1109/TPDS.2024.3497792.

[9] Eugene Furman et al. *Capacity Allocation for Clouds with Parallel Processing, Batch Arrivals, and Heterogeneous Service Requirements*. 2022. arXiv: 2209.08820 [stat.AP]. URL: https://arxiv.org/abs/2209.08820.

[10] Alexander Rumyantsev et al. "Three-level modeling of a speed-scaling supercomputer". en. In: *Annals of Operations Research* (2022). DOI: 10.1007/s10479-022-04830-0. (Visited on 06/21/2022).

[11] Mor Harchol-Balter. "Open problems in queueing theory inspired by datacenter computing". en. In: *Queueing Systems* 97.1-2 (2021), pp. 3–37. DOI: 10.1007/s11134-020-09684-6.

[12] Mark L. Huber. *Perfect Simulation*. Chapman and Hall/CRC, Jan. 2016. ISBN: 9780429165269. DOI: 10.1201/b19235.

[13] A. Golovin, A. Rumyantsev, and S. Chakravarthy. *Regenerative Exact Simulation for Multiserver Job Model*. Unpublished.

[14] Karl Sigman. "Exact Simulation of the Stationary Distribution of the FIFO M/G/c Queue". In: *Journal of Applied Probability* 48A (2011). Publisher: Applied Probability Trust, pp. 209–213. DOI: 10.1239/jap/1318940466.

[15] Wilfrid Kendall. "Geometric Ergodicity and Perfect Simulation". In: *Electronic Communications in Probability* 9.none (2004). DOI: 10.1214/ecp.v9-1117.

[16] Erol Gelenbe and Isi Mitrani. *Analysis and Synthesis of Computer Systems*. IMPERIAL COLLEGE PRESS, 2010. ISBN: 9781848163966. DOI: 10.1142/p643.

[17] Sheldon M. Ross. *Stochastic processes*. 2nd ed. Wiley series in probability and statistics. New York: Wiley, 1996. ISBN: 978-0-471-12062-9.

[18] Karl Sigman. "Stationary Marked Point Processes". In: *Springer Handbook of Engineering Statistics*. Springer London, 2006, pp. 137–152. DOI: 10.1007/978-1-84628-288-1_8.

[19] G. Y. Fletcher, H. G. Perros, and W. J. Stewart. "A queueing system where customers require a random number of servers simultaneously". In: *European Journal of Operational Research* 23.3 (1986), pp. 331–342. DOI: 10.1016/0377-2217(86)90299-7.

[20] Evsey Morozov, Alexander Rumyantsev, and Irina Peshkova. "Monotonicity and stochastic bounds for simultaneous service multiserver systems". In: *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2016 8th International Congress on*. IEEE, 2016, pp. 294–297. DOI: 10.1109/ICUMT.2016.7765374.

[21] Colin M. Ramsay. "Exact waiting time and queue size distributions for equilibrium M/G/1 queues with Pareto service". en. In: *Queueing Systems* 57.4 (2007), pp. 147–155. DOI: 10.1007/s11134-007-9052-7.