September 2014

Perfect and Nearly Perfect Sampling of Work-conserving Queues

Yaofei Xiong
The University of Western Ontario

Supervisor
Duncan J. Murdoch
The University of Western Ontario

Joint Supervisor
David A. Stanford
The University of Western Ontario

Graduate Program in Statistics and Actuarial Sciences

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

PERFECT AND NEARLY PERFECT SAMPLING OF WORK-CONSERVING QUEUES

(Thesis format: Monograph)

by

Yaofei Xiong

Graduate Program in Statistics and Actuarial Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

We present sampling-based methods to treat work-conserving queueing systems. A variety of models are studied. Besides the First Come First Served (FCFS) queues, many efforts are putted on the accumulating priority queue (APQ), where a customer accumulates priority linearly while waiting. APQs have Poisson arrivals, multi-class customers with corresponding service durations, and single or multiple servers.

Perfect sampling is an approach to draw a sample directly from the steady-state distribution of a Markov chain without explicitly solving for it. Statistical inference can be conducted without initialization bias. If an error can be tolerated within some limit, i.e. the total variation distance between the simulated draw and the stationary distribution can be bounded by a specified number, then we get a so called "nearly" perfect sampling.

Coupling from the past (CFTP) is one approach to perfect sampling, but it usually requires a bounded state space. One strategy for perfect sampling on unbounded state spaces relies on construction of a reversible dominating process. If only the dominating property is guaranteed, then regenerative method (RM) becomes an alternative choice. In the case where neither the reversibility nor dominance hold, a nearly perfect sampling method will be proposed. It is a variant of dominated CFTP that we call the CFTP Block Absorption (CFTP-BA) method.

Time-varying queues with periodic Poisson arrivals are being considered in this thesis. It has been shown that a particular limiting distribution can be obtained for each point in time in the periodic cycle. Because there are no analytical solutions in closed forms, we explore perfect (or nearly perfect) sampling of these systems.

Keywords: Perfect sampling, Nearly perfect sampling, Work-conserving queues, Priorities, Homogeneous and time-varying queues

Acknowlegements

As I have reviewed my thesis, I am somehow surprised since I feel I have gained so much knowledge and have accomplished some findings through the Ph.D. program. But I know that I cannot achieve these only by myself and I owe a lot to other people.

My deepest gratitude will be expressed to my supervisors, Dr. Murdoch and Dr. Stanford, for their excellent guidance, patience and help. Dr. Murdoch led me into the area of perfect sampling hand in hand. He helped me to understand many complicated concepts and taught me a variety of practical techniques. As a queueing theory expert, Dr. Stanford's experience and knowledge in this area made my sight broader.

I would like to thank Dr. Hao Yu and Dr. Qi-Ming He. After taking three courses instructed by Dr. Yu, I learnt quite a lot in statistics. I got to know Dr. He in the conference CanQueue2013. He gave me constructive suggestions for the improvement of my research.

I would also like to thank my parents for their endless supporting and encouraging, which have powered me since I was born.

Finally, I would like to thank my wife, Lan Zhang, for her always cheering me up and standing by me in good or bad times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents perfect (or nearly perfect) sampling of some work-conserving queueing systems with homogeneous or time-varying inputs. For the homogeneous queues, we focus on the analysis of accumulating priority queues (APQ) to demonstrate our methods concretely. For the time-varying queues, we study the quasi-birth and death processes and periodic Poisson arrival queues under the First Come First Served (FCFS) discipline.

In this chapter, background and motivation are introduced, key methods mentioned and contents of this thesis outlined.

1.1 Accumulating priority queue

The accumulating priority queue was introduced as the "time-dependent priority queue" by Kleinrock [31]. It is a queue in which customers accumulate priority as a linear function of their time in the queue: the higher the priority class of a customer, the greater the rate at which that customer accumulates priority. When the server becomes free, the customer with the highest priority accumulated to that instant, if any, is the one that is selected by the server.

Whereas Kleinrock [31] derived a set of recursive formulae for the average waiting time for the different classes, Stanford et al. [54] extended Kleinrock's analysis to derive the Laplace-Stieltjes Transform (LST) of the stationary waiting time distribution for each class in the Poisson arrival, general service duration and single-server case.

In [54], the APQ was motivated by applications in health care. Generally, patients are classified according to some acuity rating system, such as the Canadian Triage and Acuity Scale (CTAS) [11], as shown in Table 1.1 with some Key Performance Indicators (KPIs). It is not reasonable to assign absolute priorities to patients with relatively different service requirements, where a patient of lower priority classes can be overtaken many times by those of higher priority, without any priority being accrued while the patient waits with possibly deteriorating

Category	Classification	Access	Performance level (%)
1	Resuscitation	Immediate	98
2	Emergency	15min	95
3	Urgent	30min	90
4	Less urgent	60min	85
5	Not urgent	120min	80

Table 1.1: CTAS key performance indicators. Performance level (in percentage) is the compliance target for the proportion of that class's patients that need to meet that standard.

health. The APQ rectifies this weakness by allowing waiting customers to earn priority while waiting, at a rate that depends upon their priority class. In this way, low priority customers eventually earn enough credit to be served ahead of recently-arrived high priority ones.

Based on Stanford et al. [54], Sharif et al. [50] considered the multi-server case, with the additional requirement that the service durations be identically and exponentially distributed. Then numerical inversions of Laplace transforms are performed with the Gaver-Stehfest method (c.f. Abate and Whitt [3], Gaver [16] and Stehfest [55]) to calculate the probabilities of waiting times exceeding some limits. At present, no equivalent analytical result exists for more general cases, therefore sampling-based methods are strong candidates.

1.2 Perfect sampling

In Markov Chain Monte Carlo (MCMC), a common approach to generate draws from a "steady-state" distribution of a Markov chain is to sample from an arbitrary starting point, then discard the first so many draws. This discarding is done in order to remove any possible bias due to the initial conditions at the start of the simulation; such bias is sometimes referred to as "initialization bias" [37, p. 287] and the period of time discarded is referred to as the "burn in". The burn-in period is chosen to be long enough that the remaining draws are close to the steady-state distribution. Steady-state properties are estimated using the remaining draws. A practical difficulty is that for most chains the appropriate burn-in time is unknown.

Perfect sampling is an approach to draw a sample directly from the steady-state distribution without explicitly solving for it. It is also called "exact sampling", "perfect simulation" or "exact simulation". With perfect sampling, the burn-in time is not an issue.

The first well-known perfect sampling algorithm is commonly referred to as Coupling From The Past (CFTP), proposed by Propp and Wilson [45]. Actually, Asmussen et al. [9] had achieved similar results with different methods several years before, but it was prohibitively inefficient in terms of computer time as Asmussen and Glynn [6, p. 121] have mentioned.

1.2.1 Coupling from the past

Conceptually, an infinitely long run of the chain is simulated as starting in the indefinite past, so that the draw at time 0 is in steady state. The original CFTP by Propp and Wilson [45] mainly considers sampling steady-state draw for finite-state Markov chains. If one were to run coupled chains from all possible states at a finite time in the past, and if all of them result in the same output at time 0 (i.e., they coalesce by time 0), then this value will be identical to what would be achieved in any longer run, so its distribution must be the steady-state distribution.

To facilitate the implementation of algorithms, Read Once Coupling From The Past (ROCFTP) was presented by Wilson [58], where the random variables which drive the coupled Markov chains are used only once.

People might give up when encountering a long run of CFTP, and introduce "user-impatience bias" (named by Fill [15]). Fill invented a "rejection sampling" algorithm for perfect sampling. Starting from a given time in the future, firstly it performs simulation backwards from an arbitrary state and ends at time 0. Then it goes forward running chains from all possible states with the random numbers generated in the first step until the originally given time. Finally it accepts (or rejects) the state at time 0 if all chains coalesce (or do not). This method is nice since it avoids impatience bias, but it requires simulation of the time-reversal of the chain which is hard to achieve.

Wilson [57] proposed Multishift Coupling for families of location shifted distributions, which allows efficient coupling of Markov chains with continuous state spaces. This allows coalescence to be detected by the coalescence of minimal and maximal states. It is a "monotone" coupling, in the sense that the ordering of states is preserved in each transition.

"Dominating coupling" by Kendall and Møller [29] is an important extension of CFTP. We call this extended version as dominated CFTP. It enables coupling Markov chains with unbounded state spaces by reducing the number of past chains that need to be simulated. For situations with a natural (partial) ordering on the state space, simpler chains that dominate the target ones are constructed and simulated backwards from time zero. When going forward, the target chain only need to be simulated from values lying below the dominators. The construction of the reversible dominating chain is the key in this method.

1.2.2 Regenerative method

The workload or queue length of a stable queueing system can be treated as a regenerative process. The regenerative points are the instants with a customer entering an empty system. The stopping time is the length of a conventional busy cycle (a busy period followed by an idle one) as noted by Asmussen and Glynn [6, p. 112].

As shown by Sigman [52], Asmussen et al. [9] and Asmussen and Glynn [6, p. 420], we can simulate exactly from the stationary distribution of the queueing system if we can simulate exactly from the equilibrium distribution of a busy cycle length, i.e. the distribution of the residual of a randomly selected busy cycle.

The outstanding advantage of this method is quite appealing: it does not need a reversible chain. But its drawback is that the expected runtime is infinite (see Proposition 2.4.2). In practice, this algorithm will always finish in finite time, but occasionally will take a very long time to do so.

1.2.3 Nearly perfect sampling

Truly perfect sampling is hard to achieve due to high dimensions or the difficulty in reversible dominating chain construction. Fortunately, in the queueing context, when the stable queue can be treated as a stochastic process with unfinished workload (or queue length) as the variable, the high dimension issues can be avoided.

Nearly perfect sampling can be considered as an asymptotic perfect sampling with well specified distance to the target distribution. The quantitative assessment of it makes this method valuable. As shown by Asmussen and Glynn [6, p. 100], and Zeifman et al. [60], the upper bound of differences between the first moments of the transient distributed samples and the stationary ones can be controlled to be guaranteed to be within an arbitrary distance.

In this thesis, we define the difference as the total variation distance [37, p. 47] between the simulated draw and the stationary distribution. Using our CFTP Block Absorption (CFTP-BA) method (Section 3.5), we can simulate samples guaranteed to be within a 10^{-10} total variation distance of the stationary distribution in a few seconds of computing time.

1.3 Time-varying queues

Time-varying queueing models are more realistic, but they are not usually mathematically tractable (Ross [49, p. 697]). As noted by Margolius [41], computational methods and approximation techniques involved in time-varying queueing problems have long been regarded as challenging. The time-varying ingredients can exist in the arrival processes, service durations, or the number of servers, as mentioned by Alfa and Margolius [4].

Generally, it is acceptable that the time-dependent stochastic processes take some periodic patterns. As for the periodic Poisson arrival single-server queue with general service duration, Hasofer [22] showed that the LST of its virtual waiting time (i.e. unfinished workload) is asymptotically periodic in time. Harrison and Lemoine [21] proved that the virtual waiting

time at a given time does have a limiting distribution and has the same period length as the arrival rate does.

Asmussen and Thorisson [8] extended the context to more general cases, where the inter-arrival times and service durations both depend on the arrival instant with some periodic pattern. They proved that with more conditions (such as Harris ergodicity of the phase parameter which the inter-arrival time and service duration depend on), the virtual waiting time and queue length also have time dependent limiting distributions in periodic patterns. Due to the complexity of time-varying systems, only asymptotic solutions have been developed, and this has happened gradually over recent decades.

Time-varying quasi-birth and death processes have been frequently studied for the transient or periodic solutions. By assuming some state (generally idle) at time 0, Zhang [61] and Margolius [40] figured out the transient distributions of queue length in the single-server and multi-server cases respectively. Zeifman et al. [60] approximated the limiting mean value (expected queue length at some given time) of the single-server model with the transient distribution by restricting their difference to some controllable extent. The asymptotic periodic solutions for single-server and multi-server models were achieved in Margolius [41], where distributions and moments were presented in terms of integral equations.

In this thesis, queueing systems are generally presumed to be homogeneous ones unless they are specifically pointed out as time-varying.

1.4 Problems to be solved

Analytically intractable models, such as Poisson arrival, multi-server multi-class APQs with differently distributed service durations, and time-varying APQs with periodic Poisson arrivals, are good candidates to apply perfect (or nearly perfect) sampling.

For partly solved models (e.g. results of the time-varying queues with periodic Poisson arrivals by Lemoine [36] only present the moments of some statistics such as workload and waiting time), the perfect sampling method provides direct solutions for the probability mass of the queue length and tail probabilities in a simulation based way without having to approximate based upon moment-based methods (c.f. Provost et al. [46]).

For problems with solutions in LST forms (like Hasofer [22], Stanford et al. [54] and Sharif et al. [50]), applications of these methods are also validated in the sense that they provide alternative and comparable solutions to the numerically inverted LST ones. As noted by Abate and Whitt [3], the commonly used inversion, by Gaver [16] and Stehfest [55], only has limited accuracy, restricted by the number of transform evaluations and computer system precision limits.

In the work-conserving context, these methods are adaptive automatically to any priority disciplines specified, because they do not affect the computation of the tail of the infinitely long run. More specifically, as for the work-conserving single server queue, the unfinished workload path stays invariant no matter what priority disciplines are applied.

1.5 Outline of this thesis

As noted earlier, the dominated CFTP achieves perfect sampling of the Markov chains with unbounded state within finite expected runtime, so it is the the preferred choice in what follows. But when the reversible dominating process is hard to construct, we resort to the regenerative method or nearly perfect sampling by CFTP Block Absorption (CFTP-BA).

In Chapter 2, notations and models are specified, and related existing algorithms are introduced as components for addressing new problems.

These algorithms are: 1) CFTP related (original CFTP, multishift coupler, and reversion of single-server queue with Poisson arrivals); 2) other perfect sampling methods (regenerative method and a special case for general single-server queue); 3) miscellaneous ones (ordinary simulation of multi-server queue with Random Assignment (RA), time-varying Poisson process simulation, and Gaver-Stehfest algorithm for numerical inverse of LST).

Chapter 3 deals with homogeneous queues with single or multiple servers. After applying the CFTP to a single-server queue with heavy tail inter-arrival time and service duration, we go to APQs. Perfect sampling methods are applied to various queueing models with Poisson arrivals and general service durations. When the service distributions differ among different classes, the reordering of service durations will affect the distribution of the busy period in the multi-server case. The new method we call CFTP-BA will be introduced and applied.

Chapter 4 explores time-varying queueing systems. Periodic Poisson arrivals are assumed, and the service durations could be periodically time-dependent exponential or homogeneous general ones. We focus on the FCFS discipline and briefly describe quick extensions to some APQ models.

Results and contributions are summarized in Chapter 5. Some related new topics will be pointed out as possible future work.

Chapter 2

Preliminaries

2.1 Notation and terminology

For the sake of clearness and consistency, abbreviations and miscellaneous mathematical notations are shown in Tables 2.1 and 2.2 respectively.

For different queueing systems, Kendall's notation (c.f. [28]) is used as the standard classifier. Since we assume an unlimited waiting room and infinitely large population of customers, and specify the discipline additionally, the three-part code $(a/b/c)$ is enough. When it is not explicitly specified, it is presumed there are infinite waiting room and population of customers. The first letter indicates inter-arrival time distribution, the second one service duration distribution, and the third one the number of servers. Conventionally, "M" stands for the exponential distribution, and "G" for an unspecified "general" distribution.

In this thesis, it is assumed that the inter-arrival times are independent (corresponding to notation "GI"), service durations independent, and the service durations are also independent of the inter-arrival times.

In the priority queueing systems, in the light of Stanford [53], we introduce notation "Σ^K" pointing out that the arrival process is a superposition of K independent streams. If the classes of customers might have different distributions of service durations, subscript K is added to the service code. E.g. $\Sigma^K\mathrm{M}/\mathrm{M}_K/1$ stands for a single-server priority queue with K (≥ 2) classes of customers arriving in Poisson processes, and each class has its own exponential service duration distribution.

As for the time-varying queues, as noted in some recent papers (c.f. Margolius [40] and Zeifman et al. [60]), subscript t implies that the inter-arrival time or service duration's distributions are time dependent. For instance, the notation $\mathrm{M}_t/\mathrm{M}_t/1$ indicates that it is a single-server queue with time-varying Poisson arrival and the service duration is exponential with time varying rate.

APQ	Accumulating priority queue
c.d.f.	Cumulative distribution function
CFTP	Coupling from the past
CFTP-BA	CFTP with block absorption
ECM	Exponential change of measure
e.c.d.f	Empirical cumulative distribution function
FCFS	First come first served
i.i.d.	Independent and identically distributed
LST	Laplace-Stieltjes Transform
p.d.f.	Probability density function
p.g.f.	Probability generating function
PS	Processor sharing
RA	Random assignment
RM	Regenerative method
r.v.	Random variable
WCQ	Work-Conserving Queue

Table 2.1: Abbreviations

$\mathbb{Z}$	Integers: $0, \pm 1, \pm 2, \ldots$
$\mathbb{N}$	Positive integers: 1,2,...
$\mathbb{R}$	Real numbers
$\overset{D}{=}$	Identically distributed
$\overset{D}{\geq}$	Large or equal statistically
$=_{so}$	Stochastically equal
$\geq_{so}$	Stochastically larger or equal
$\perp$	Independent
$\exists$	Exists
$\forall$	For all
$\ni$	Such that
$\lceil x \rceil$	The smallest integer which is no less than x
$\lfloor x \rfloor$	The largest integer which is no greater than x
$(x)^+$	The non-negative truncated value of x, i.e. x if $x > 0$ or 0 if $x \leq 0$
Exp(λ)	Exponential distribution with rate of λ
Geom(p)	Geometric distribution with success probability of p,
Poi(λ)	Poisson distribution with arrival rate of λ
Unif(0, 1)	Uniform distribution on (0, 1)
NB(r, p)	Negative binomial distribution with number of successes r and success probability p

Table 2.2: Miscellaneous mathematical notations

Busy period in multi-server queues The busy period in multi-server queues is defined as a duration started at the arrival instant when the arriving customer finds an empty system, and after that for the first time terminated at the departure instant, when the departing customer leaves behind no busy servers. See Wiens [56] and Ghahramani [17]. The latter called it "partial busy period".

Totally idle period in multi-server queues It is the duration in the multi-server system when all servers are idle.

Reversibility A stochastic process $X(t)$ is reversible [27, p. 5] if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$ for all $t_1, t_2, \ldots, t_n, \tau \in \mathbb{R}$.

In a word, when going forward or backwards along this process in time, what we see are statistically equivalent. So we also call it time reversible.

Light/Heavy tail distributions We define light tail distribution as those which decay at an exponential rate or faster (c.f. Asmussen and Glynn [6, p. 163]). In queueing studies, usually the distributions of interest have positive support $(0, \infty)$. So a distribution $G(\cdot)$ of light tail requires that there exists $\epsilon > 0$ such that

$$\int_0^\infty e^{\epsilon x} dG(x) < \infty.$$

We consider heavy tail distributions as those which have super-exponential tails, i.e. $\int_0^\infty e^{\epsilon x} dG(x) = \infty$ for all $\epsilon > 0$ [5, p. 412].

Total variation distance The total variation distance between two probability distributions ν_1 and ν_2 on Ω is defined by

$$\|\nu_1 - \nu_2\|_{TV} = \max_{E \subset \Omega} |\nu_1(E) - \nu_2(E)|.$$

It can be computed as

$$\|\nu_1 - \nu_2\|_{TV} = \frac{1}{2} \sum_{x \in \Omega} |\nu_1(x) - \nu_2(x)|. \tag{2.1}$$

A brief explanation is presented as follows. Let $\mathcal{E} = \{x : \nu_1(x) \geq \nu_2(x)\}$, then $\|\nu_1 - \nu_2\|_{TV} = \sum_{x \in \mathcal{E}} (\nu_1(x) - \nu_2(x))$. Please note that

$$\sum_{x \in \mathcal{E}} (\nu_1(x) - \nu_2(x)) = \sum_{x \in \mathcal{E}^c} (\nu_2(x) - \nu_1(x)) = \frac{1}{2} \sum_{x \in \Omega} |\nu_1(x) - \nu_2(x)|,$$

since $\nu_1(\Omega) = \nu_2(\Omega) = 1$. Therefore equation (2.1) holds.

For more details please refer to [37, p. 47-48].

2.2 Model specifications

Non-preemptive work-conserving queue We define a non-preemptive work-conserving queue discipline (NPWCQ) to be one in which the work requirements of customers are unaltered by the passage of time and the server never idles when there is work to be done. Customers that enter service remain in service until completion. This definition adds the lack of preemption to the definitions given by Gross and Harris [19, p. 299], and Kleinrock [33, p. 113]. This means, in particular, that customers do not renege. When there exists an idle server, a customer's service startes upon arrival. Otherwise, the customer joins a pooled queue and will be selected to go into service according to some discipline when any server becomes available.

For writing convenience, in this thesis we use "work-conserving queue" (WCQ) to stand for NPWCQ.

2.2.1 Accumulating priority queue

This discipline was first proposed by Kleinrock [31] and was termed as "time-dependent priority queue". According to Stanford et al. [54], the specification of the APQ is described below.

Assume there are $K \in \mathbb{N}$ ($K \geq 2$) classes of customers, and one or c (≥ 2) servers in the system. Each class of customers arrives independently in a Poisson process with rate λ_i, $i = 1, \ldots, K$.

For class i customers, the priority accumulates linearly at rate b_i, i.e. if a customer of class i arrived at time t and is still in the system at time t', then its priority at time t' is $b_i(t' - t)$, and $b_1 > b_2 > \cdots > b_K$. When any server is available, the next customer to be served is the one with the greatest priority at that instant. This is a non-preemptive system.

Let A be the inter-arrival time, whose c.d.f. is $\mathcal{A}(x)$, of two successive customers, and $A^{(i)}$ be that of class i customers. According to the aggregation and branching property (c.f. Conway et al. [13, p. 143]) of the Poisson process, it follows

$$A \sim \text{Exp}(\lambda), A^{(i)} \sim \text{Exp}(\lambda_i), \text{ and } A = \min\{A^{(i)}, i = 1, \ldots, K\}$$

where $\lambda = \sum_{i=1}^{K} \lambda_i$. And a customer is classified as class i with probability of $\frac{\lambda_i}{\lambda}$.

Let $B^{(i)}$ be the service duration of class i customers with c.d.f. $G_i(x)$, and B be the generic service duration with c.d.f. $G(x)$ under the FCFS discipline, then G is a mixture of G_i, i.e.

$$G(x) = \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} G_i(x). \tag{2.2}$$

To ensure some important statistics (e.g. the first moment of the stationary residual length

of a randomly selected busy period of the M/G/1 FCFS queue) do exist, we assume

$$\mathbb{E}\left(\left(B^{(i)}\right)^2\right) < \infty.$$

It follows that $\mathbb{E}\left(B^{(i)}\right) < \infty$, and $\mathbb{E}\left(B^2\right) < \infty$. Denote the corresponding service rates as

$$\mu_i = \frac{1}{\mathbb{E}\left(B^{(i)}\right)}, \text{ and } \mu = \frac{1}{\mathbb{E}(B)}.$$

In the single-server scenario, the occupancies are

$$\rho_i = \frac{\lambda_i}{\mu_i}, \text{ and } \rho = \frac{\lambda}{\mu}.$$

In the multi-server scenario

$$\rho_i = \frac{\lambda_i}{c\mu_i}, \text{ and } \rho = \frac{\lambda}{c\mu}.$$

It is easy to verify that

$$\rho = \sum_{i=1}^{K} \rho_i,$$

since the mean service duration is the weighted average of those of all classes

$$\frac{1}{\mu} = \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} \frac{1}{\mu_i} \Rightarrow \frac{\lambda}{\mu} = \sum_{i=1}^{K} \frac{\lambda_i}{\mu_i} \Leftrightarrow \rho = \sum_{i=1}^{K} \rho_i.$$

To ensure the system is stable [32, p. 19], it must be assumed that

$$\rho < 1,$$

which guarantees that the system empties occasionally, with probability 1.

2.2.2 Time-varying queues

In this thesis we also consider some time-varying queues, specifically those with periodic arrival and service processes. Without loss of generality, we assume the period length is 1.

According to Asmussen and Thorisson [8], the periodic single-server queue can be defined as follows: at the arrival instant of the n^{th} customer, say at time t, the service duration B_n and the inter-arrival time, A_n, to the next arrival, are drawn according to distributions with c.d.f. G_θ and $\mathcal{A}_\theta$, which depend on the phase $\theta = (t \bmod 1)$ at the arrival instant.

Model to be treated	Description
GI/G/1 FCFS queue	Inter-arrival time and service duration are both heavy tailed.
Σ^KM/G$_K$/1 and Σ^KM/G$_K$/c APQs	Poisson arrivals, K classes customers with corresponding service duration distributions.
M$_t$/M$_t$/1 and M$_t$/M$_t$/c FCFS queues	Periodic Poisson arrival and time-dependent exponential service rate.
M$_t$/G/1, M$_t$/G$_K$/1 and M$_t$/G/c FCFS and APQs	Periodic Poisson arrival and general service duration distributions.

Table 2.3: Queueing models to be treated with perfect (or nearly perfect) sampling

For the time-varying quasi-birth and death processes (denoted as M$_t$/M$_t$/1 or M$_t$/M$_t$/c FCFS queues), they have time-dependent Poisson arrival rates and "time-dependent exponential service rates" [40]. Let $\lambda(t) \geq 0$ and $\mu(t) \geq 0$ be the arrival and service rates, which have period length of 1, i.e.

$$\lambda(t) = \lambda(t+1) \text{ and } \mu(t) = \mu(t+1), \forall t \in \mathbb{R}.$$

Both $\lambda(t) > 0$ and $\mu(t) > 0$ except at discrete points, so their integrals are strictly increasing. Then

$$\begin{aligned} \mathcal{A}_{\theta_1}(x) &= 1 - e^{-\int_0^x \lambda(\theta_1+s)ds}, \\ G_{\theta_2}(x) &= 1 - e^{-\int_0^x \mu(\theta_2+s)ds}, \end{aligned}$$

where θ_1 is the instant of arrival, and θ_2 the instant of entry into service. Under the FCFS discipline, θ_2 can be determined with the unfinished workload seen at θ_1.

In the priority systems, we assume all classes of customers arrive as independent periodic Poisson processes with

$$\lambda(t) = \sum_{i=1}^{K} \lambda_i(t), \ \forall t \in \mathbb{R}.$$

Service durations have homogeneous distributions, because the entry into service times are affected by the priority discipline, it is hard to determine them at the arrival instants. So these models are denoted as Σ^KM$_t$/G$_K$/1 or Σ^KM$_t$/G/c.

2.2.3 Queueing models to be treated

With notations and models being specified above, we summarize the models to be treated with perfect (or nearly perfect) sampling methods in Table 2.3.

2.3 CFTP related algorithms

2.3.1 CFTP

The CFTP algorithm was introduced by Propp and Wilson [45]. It allows perfect sampling of an ergodic Markov chain. It can be applied to bounded and continuous state space chains.

For the refined algorithms and brief proofs of CFTP, see Murdoch and Takahara [43] and Asmussen and Glynn [6, p. 120].

We will describe it for the case of a finite state space labelled as $1, 2, \ldots, n$. Denote by X_t the ergodic Markov chain. Suppose that it can be simulated using a recursive formulation

$$X_{t+1} = \phi(X_t, U_{t+1}),\ t \in \mathbb{Z}, \tag{2.3}$$

where U_{t+1} are i.i.d. from some known distribution, and ϕ is a deterministic function.

In the CFTP context, let $X_t^{(\tau_m, j)}$ be the Markov chain starting from time τ_m with state j, where $\tau_m < 0$, $m = 0, 1, \ldots$, and $j = 1, 2, \ldots, n$. We will usually set $\tau_m = -2^m T_0$, where $T_0 \in \mathbb{N}$ is a constant.

The algorithm can be performed in the following way:

(1) Run the initial trial (i.e. $m = 0$ and $\tau_0 = -T_0$) to detect the coalescence.

Generate i.i.d. U_t ($t = \tau_0 + 1, \tau_0 + 2, \ldots, 0$). Then update $X_t^{(\tau_0, j)}$ ($j = 1, 2, \ldots, n$) with formula (2.3) and these random numbers.

If $X_0^{(\tau_0, j)} = X_0^{(\tau_0)}, \forall j = 1, 2, \ldots, n$, i.e. they have coalesced by time 0, then output $X_0^{(\tau_0)}$ as the steady-state draw. Otherwise, go to step (2).

(2) Conduct extra trials (i.e. $m \geq 1$ and $\tau_m = -2^m T_0$) starting with $m = 1$:

Generate extra i.i.d. U_t ($t = \tau_m + 1, \tau_m + 2, \ldots, \tau_{m-1}$). Then update $X_t^{(\tau_m, j)}$ ($j = 1, 2, \ldots, n$) with formula (2.3) and U_t ($t = \tau_m + 1, \tau_m + 2, \ldots, 0$).

If $X_0^{(\tau_m, j)} = X_0^{(\tau_m)}, \forall j = 1, 2, \ldots, n$, then stop repeating and output $X_0^{(\tau_m)}$ as the steady-state draw. Or else increase m by 1 and repeat this step.

Remark

(1) In the extra trials of detecting coalescence, the random variables generated in previous trials must be reused.

(2) The choice of T_0 depends on the characteristics of the system. E.g. in the queueing system, if the occupancy rate is close to 1, which means the stationary queue length is likely to be large

and it tends to take a long run to return zero, so we prefer a relatively large T_0, which allows a greater chance of coalescence to occur.

(3) τ_m could take $-2mT_0$ ($m \geq 1$), $-2^m T_0$, or other values. The geometric scheme accelerates coalescence in the extra trials.

(4) The runtime is finite due to the "geometric trial argument" (see proposition 8.1 by Asmussen and Glynn [6, p. 122]).

2.3.2 Multishift coupling

Before explaining this algorithm, we state the concept of monotonicity (see Murdoch and Takahara [43] and Propp and Wilson [45]). An update function is said to be monotonic if it preserves the partial order in the state space, i.e. $X \preceq Y$ implies $\phi(X, u) \preceq \phi(Y, u)$ for all u. With monotonicity, only maximal and minimal elements need to be followed, as all others are sandwiched between them.

Here "$\preceq$" represents the partial order relation. If X and Y are scalar values, then the partial order could be the usual numerical order. When they are vectors, we can define a componentwise partial order as $X \preceq Y$ if $X^{(i)} \leq Y^{(i)}$ for all i.

The choice of update function $\phi(\cdot, \cdot)$ is crucial. It must be chosen so that updates coalesce and coalescence can be detected, which is not always easy. For example, we need to simulate shifted exponential completion instants W when simulating the completion of service for an individual who starts service at time $s(X_t)$. A simple choice would be to simulate $E \sim \text{Exp}(\mu)$, whose c.d.f. has the form $1 - e^{-\mu x}$, $x > 0$, and set $W(X_t) = s(X_t) + E$, but then different $s(X_t)$ values would always lead to different $W(X_t)$ values.

Wilson [57] proposed Multishift Coupling for families of location shifted distributions, and we can use this coupling to handle unimodal distributions. Continuing with the example above, for $s(X_t) = 0$ we set $W = E$, but use the following construction (see Figure 2.1) for other values. After obtaining E, we sample $U \sim \text{Unif}(0, 1)$ and multiply it by the density at E to obtain $D = U\mu \exp(-\mu E)$ and the point $C_0 = (E, D)$. We then compute H by setting $D = \mu \exp(-\mu H)$, and replicate C_0 as $C_n = (E + nH, D), n \in \mathbb{Z}$ (shown as asterisks in the plot). For any value of $s(X_t)$, exactly one of these points lies under the shifted density; we use that point's horizontal coordinate as $W(X_t)$.

By construction, $W(X_t)$ takes on a discrete lattice of values as $s(X_t)$ varies. This is a monotone coupling, i.e. it preserves monotonicity, in the sense that the ordering of $s(X_t)$ values is preserved in the corresponding $W(X_t)$ values, and our simulations from multiple starting values will result in coalescence when $s(X_t)$ falls in a sufficiently small interval. E.g. as shown in Figure 2.1, for any $s(X_t)$ lying between C_1 and C_2, they are updated to $W(X_t)$.

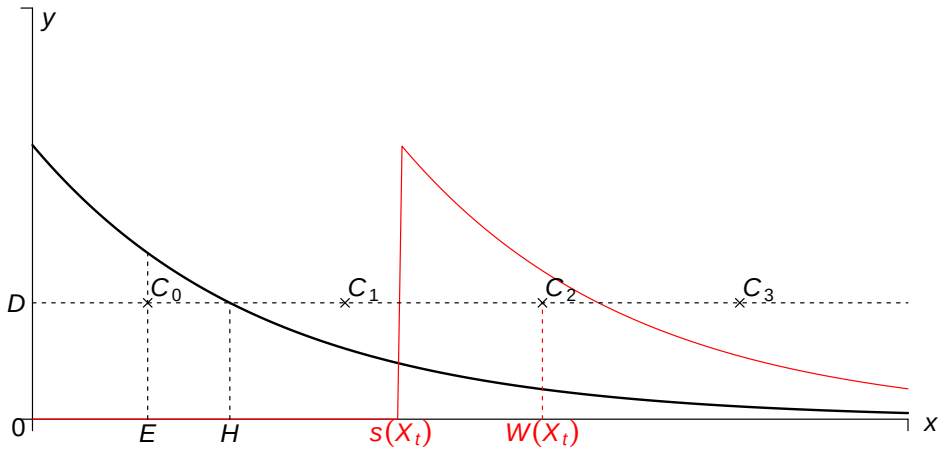

Figure 2.1: An illustration of multishift coupling. The solid black line is the standard Exp(μ) density; the red line is the density after shifting to an origin of $s(X_t)$. The values E and D are selected at random as described in the text; points C_i and the value $W(X_t)$ are derived.

With this method, we can perform coupling in a continuous state space. Uncountably many chains will coalesce to a finite number of different states at the first transition.

2.3.3 Dominated CFTP

As mentioned in Section 2.3.1, ordinary CFTP is only easily applied to bounded state chains. However, in queueing models, the state space (e.g. unfinished workload or queue length) is usually unbounded, so we need to upgrade our method. The dominated CFTP was introduced by Kendall and Møller [29]. Its basic idea is to reduce the number of chains to be simulated. We want to sample a steady-state draw from $\{X_t\}_{t\in\mathbb{R}}$, but it is too complex to implement. Suppose we can construct a dominating process $\{Y_t\}_{t\in\mathbb{R}}$, which dominates our target process in the following sense. Let $\preceq$ be a partial order on the common state space of X_t and Y_t. We say Y_t dominates X_t if for any t_0 where $X_{t_0} \preceq Y_{t_0}$ we have $X_t \preceq Y_t$ for all $t \geq t_0$ with probability one, i.e. sample paths of X_t are caught below sample paths of Y_t. Assume we know how to sample from the dominating process and achieve the backward simulation. So we can use it as an upper bound to conduct the ordinary CFTP, and the unbounded problem is solved.

It is quite appealing to apply dominated CFTP for queueing systems, because in many stable queues the process state can be represented as a scalar and the empty state can be reached

within finite time.

Proposition 2.3.1 *Suppose we have a coupled simulation of two stable queues, denoted by* $\{X_t\}_{t\in\mathbb{R}}$ *and* $\{Y_t\}_{t\in\mathbb{R}}$*. We assume the following:*

1. *Both are real-valued, with 0 as a minimal state.*
2. Y_0 *is a draw from the steady-state distribution of* Y_t*.*
3. *We can simulate* Y_t *backwards in time, and find an instant* $T^a \leq 0$ *at which* $Y_{T^a} = 0$*.*
4. Y_t *dominates* X_t *in the usual ordering for real numbers.*

Then in the coupled simulation of X_t *started from* $X_{T^a} = 0$*,* X_0 *will be a draw from the steady-state distribution of* X_t*.*

Proof Start both chains in state 0 at time $t_0 < 0$. Then $Y_t \geq X_t$ for all $t \geq t_0$ by dominance. Let $t_0 \to -\infty$; then both X_0 and Y_0 tend to their steady-state distributions. The coupling only allows one possible path for X_t on $[T^a, 0]$, so X_0 as constructed above must be a steady-state draw. □

Figure 2.2 illustrates the dominated CFTP in a queueing system.

2.3.4 Backward simulation of M/G/1 FCFS queue

As mentioned before, the key of dominated CFTP is to find a dominating chain and we are able to simulate the reversal. Generally, the reversibility is harder to achieve than dominance. One feasible way is to construct a coupled time reversible chain.

It is well-known that the output process of an ergodic M/M/1 FCFS queue is time reversible (Ross [49, p. 399]), since it is a birth and death process, which is time reversible. But the M/G/1 FCFS queue is not.

Sigman [51] presented a method to simulate the M/G/1 FCFS queue backwards, based on Theorem 5.7.6 by Ross [48, p. 280]. The coupled M/G/1 Processor Sharing (PS) queue was introduced, whose output is also a Poisson process with the same rate as the arrivals.

Solutions of M/G/1 FCFS queue

Recall that, in the M/G/1 FCFS queue, according to the Pollaczek-Khintchine formula (see Kleinrock [32, p. 200]), the stationary unfinished workload has LST as

$$\widetilde{W}(s) = \frac{s(1-\rho)}{s - \lambda + \lambda\widetilde{B}(s)},$$

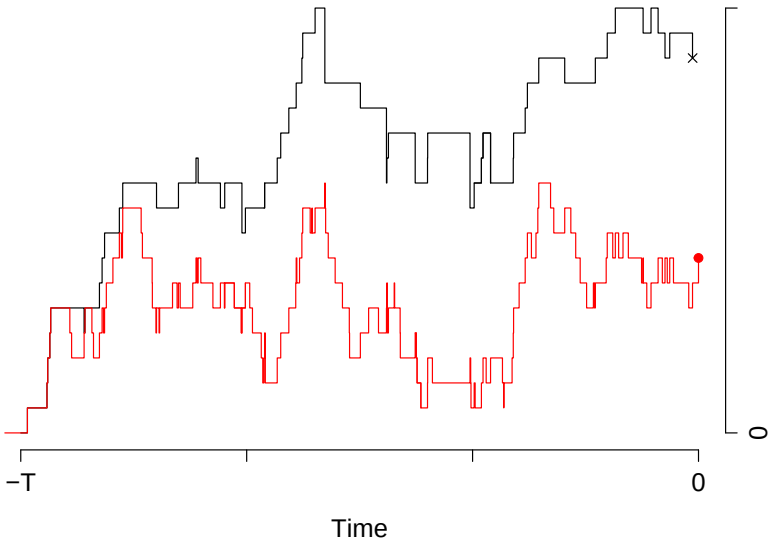

Figure 2.2: An illustration of dominated CFTP in a queueing system. These paths are the number of customers in the system. The black one is the dominator and the cross indicates a stationary draw of it at time 0. The red path belongs to the target chain and the point at time 0 is outputted as the steady-state draw of the system of interest.

where $\rho = \lambda/\mu$ is the occupancy rate, and $\widetilde{B}(s)$ the LST of service duration B. It can be written as

$$\widetilde{W}(s) = \frac{1-\rho}{1-\rho\left[\frac{1-\widetilde{B}(s)}{s/\mu}\right]} = \frac{1-\rho}{1-\rho\widetilde{B}^*(s)},$$

where

$$\widetilde{B}^*(s) = \frac{1-\widetilde{B}(s)}{s/\mu}. \tag{2.4}$$

Equation (2.4) is the LST of the equilibrium distribution of the service duration.

It is clear that

$$\frac{p}{1-(1-p)z}$$

is the p.g.f. of Geometric distribution with success probability of p. Therefore the stationary unfinished workload is a compound Geometric distribution, and it can be represented as

$$W = \sum_{i=1}^{Q} Y_i, \tag{2.5}$$

where $Q \sim \text{Geom}(1-\rho)$, $Q = 0, 1, \ldots$, and Y_i are governed by the equilibrium distribution of

the service duration and they are i.i.d.'s. If $Q = 0$, then $W = 0$.

Let T be the length of busy period of M/G/1 queue, and $\widetilde{T}(s)$ its LST. It is well known (e.g. Kleinrock [32, p. 213-214]) that

$$\widetilde{T}(s) = \widetilde{B}\Big(s + \lambda - \lambda\widetilde{T}(s)\Big), \tag{2.6}$$

$$\mathbb{E}(T) = \frac{\mathbb{E}(B)}{1-\rho}, \text{ and } \mathbb{E}(T^2) = \frac{\mathbb{E}(B^2)}{(1-\rho)^3}. \tag{2.7}$$

Coupled processor sharing queue

The Processor Sharing (PS) discipline referred in this thesis is the round-robin scheduling algorithm (Kleinrock [33, p. 166]), with all quanta of the service capacity shrinking to zero. It is a single-server system. Any customer's service starts immediately at the arriving instant, and all of them sojourning in the system share the capacity of the server equally.

The M/G/1 PS model and the M/G/1 FCFS queue are coupled in the way that they are fed with the same arrival instants and service requirements. Since workload in a single-server queue is invariant under all work-conserving disciplines, the sample paths of unfinished workload in the coupled PS model and the FCFS queue are exactly the same.

Let $Q(t)$ be the number of customers in the PS model, and $Y_1(t), \ldots, Y_{Q(t)}(t)$ the corresponding completed (or unfinished) workloads of the customers. It is shown by Ross [48, p. 280] that it has stationary distribution in the form as

$$(Q, Y_1, \ldots, Y_Q),$$

where Q and Y_i are the same as those in equation (2.5), and its departure process is also a Poisson process with rate of the arrival one (λ). Define $\vec{Y}(t) = (Y_{(1)}(t), \ldots, Y_{(Q(t))}(t))$ as the ascending ordered vector of completed workloads, then $\{Q(t), \vec{Y}(t)\}$ is time reversible. When looking forward, they are completed workloads. If looking backwards in time, they become unfinished workloads, as illustrated in Figure 2.3.

Randomly selected service duration

For a stationary M/G/1 PS queue, we check its state at time 0. Due to stationarity, the state at time 0 has the same distribution as the state at a randomly selected point in time, assuming the selection is independent of the process. Let X denote the length of a selected service duration, Y the remaining time and Z the age.

Let $\overline{G}(x)$ be the tail probability and $g(x)$ the p.d.f. (assuming it exists) of the service dura-

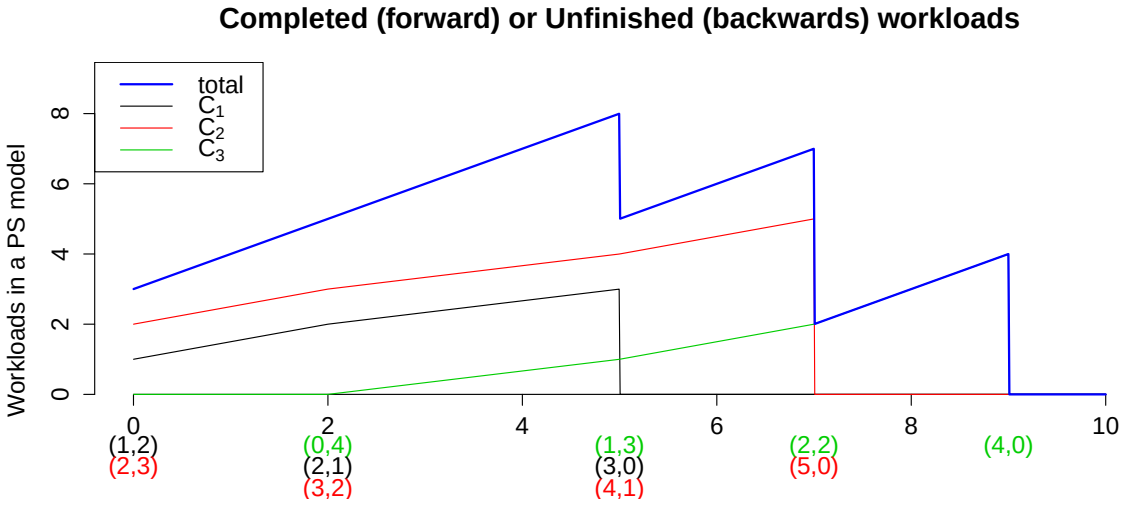

Figure 2.3: Illustration of the time reversibility of the M/G/1 PS model. Denote by (Completed, Unfinished) the workload pair. When $t = 0$ there are 2 customers (C_1 and C_2) with the pairs of (1,2) and (2,3). A new arrival (C_3) at $t = 2$ having workload pair (0,4).

tion. As shown by Kleinrock [32, p. 171-172], the p.d.f.'s of X and Y are

$$\begin{aligned} f_X(x) &= \mu x g(x) \\ f_Y(y) &= \mu \overline{G}(y). \end{aligned} \tag{2.8}$$

Let $U \sim \text{Unif}(0, 1)$ be independent of X, then due to the randomness of the inspection time, we have

$$Y = UX \text{ and } Z = (1 - U)X. \tag{2.9}$$

Because U and $1 - U$ both have standard uniform distributions, Y and Z are identically distributed. But note that they are not independent. Recall that Y represents the remaining (or unfinished) workload, and Z the completed workload. Their identical distributions support the time reversibility of the M/G/1 PS model.

Sigman [51] calls the distribution of X the spread distribution and Y having equilibrium distribution of the service duration (denoted by B). Their c.d.f.'s are denoted as

$$\begin{aligned} F_X(x) &= H(x) = 1 - \mu x \overline{G}(x) - \overline{G}_e(x), \\ F_Y(y) &= G_e(y) = \mu \int_0^y \overline{G}(s) ds, \end{aligned} \tag{2.10}$$

where $\overline{G}_e(x) = 1 - G_e(x)$. Equation (2.10) provides a more general form than equation (2.8) does.

Let T^e be the residual of a randomly selected M/G/1 busy period. Then its p.d.f. is $f_{T^e}(x) = \overline{F}_T(x)/\mathbb{E}(T)$, where $F_T(x)$ is the c.d.f. of the busy period length. So we have

$$\mathbb{E}(T^e) = \int_0^\infty x\frac{\overline{F}_T(x)}{\mathbb{E}(T)}dx = \frac{\mathbb{E}(T^2)}{2\mathbb{E}(T)} = \frac{\mathbb{E}(B^2)}{2\mathbb{E}(B)(1-\rho)^2}, \tag{2.11}$$

by using results in equation (2.7). To ensure $\mathbb{E}(T^e) < \infty$, it requires that $\mathbb{E}(B^2) < \infty$. It is one of the reasons why the assumption is made for this thesis. This is not generally considered a strict condition.

Algorithm for backward simulation of M/G/1 FCFS queue

Based on the results shown above, the algorithm can be described as follows. It implements the descriptions in Step 1 of Algorithm 1.1 by Sigman [51]. Assume the stationary busy period of a M/G/1 FCFS queue starting at time $T^a \leq 0$.

(1) Generate a r.v. $Q \sim \text{Geom}(1-\rho)$. If $Q = 0$, then return $T^a = 0$. Or else, go to step (2).

(2) Simulate forward in time the M/G/1 PS model.

Let $\vec{\tau}$ and $\vec{\beta}$ be vectors with variable lengths for the storage of departure instants and associated service durations, with initial length of zero.

Let $t_i > 0, i = 1, 2, \ldots$ denote the event (arrival or departure) instants, with $t_0 = 0$, and $Q_i, i = 1, 2, \ldots$, the number of customers in the system at t_i+, with $Q_0 = Q$.

Denote by $\vec{Y}(i) = (Y_1(t_i+), \ldots, Y_{Q_i}(t_i+)), i = 1, 2, \ldots$, the residual service requirements of customers in the system just after the instant of i^{th} event, with $\vec{Y}(0) = (Y_1(0+), \ldots, Y_Q(0+))$. Each component in $\vec{Y}(0)$ and $\vec{Y}(i), i = 1, 2, \ldots$ has its associated invariant service requirement.

Denote by a_i the time to next arrival event, and d_i to next departure event starting from t_i+, $i = 0, 1, 2, \ldots$.

- Initializations at time 0+.

  Based on equations (2.10) and (2.9), we have

$$Y_k(0+) = U_k X_k, k = 1, \ldots, Q,$$

  where $U_k \sim \text{Unif}(0, 1)$ are i.i.d., and X_k also i.i.d. with its c.d.f. defined in equation (2.10). The service requirement associated with this entry is X_k.

The time to next arrival $a_0 = E$, where $E \sim \text{Exp}(\lambda)$ is generated independently for each new arrival. Time to next departure

$$d_0 = \min_k \{QY_k(0+), k = 1, \ldots, Q\}.$$

– When $Q_i > 0$, do the follows for $i = 0, 1, \ldots$.

If $d_i < a_i$ then the next event is a departure. The updated values are as follows:

$$\begin{aligned}
t_{i+1} &= t_i + d_i; \\
Y_k(t_{i+1}+) &= Y_k(t_i+) - d_i/Q_i, k = 1, \ldots, Q_i, \\
&\quad \text{add } t_{i+1} \text{ to } \vec{\tau}, \\
&\quad \text{and as for the entry of value 0, add the associated} \\
&\quad \text{service requirement to } \vec{\beta}, \text{ then delete this entry;} \\
Q_{i+1} &= Q_i - 1; \\
d_{i+1} &= \min_k \{Q_{i+1} Y_k(t_{i+1}+), k = 1, \ldots, Q_{i+1}\}; \\
a_{i+1} &= a_i - d_i.
\end{aligned}$$

Otherwise, the next event is an arrival. Then update as

$$\begin{aligned}
t_{i+1} &= t_i + a_i; \\
Q_{i+1} &= Q_i + 1; \text{ (add a new entry)} \\
Y_k(t_{i+1}+) &= Y_k(t_i+) - a_i/Q_i, k = 1, \ldots, Q_i, \\
&\quad Y_{Q_{i+1}} \sim G(\cdot), \text{ is generated independently,} \\
&\quad \text{and associated with this entry as a service requirement;} \\
d_{i+1} &= \min_k \{Q_{i+1} Y_k(t_{i+1}+), k = 1, \ldots, Q_{i+1}\}; \\
a_{i+1} &\sim \text{Exp}(\lambda) \text{ is generated independently.}
\end{aligned}$$

Assume there are N components in $\vec{\tau}$, i.e. N departures have been generated. Since the departure events were added chronologically, the components in $\vec{\tau}$ satisfy $\tau_1 < \tau_2 < \ldots < \tau_N$, and the corresponding service requirements are $\beta_1, \beta_2, \ldots, \beta_N$ which were recorded in $\vec{\beta}$.

Output $T^a = -\tau_N$, and $(-\tau_N, -\tau_{N-1}, \ldots, -\tau_1)$ as the arrival instants of the stationary busy period's age which ends at time 0. The corresponding service durations are $(\beta_N, \beta_{N-1}, \ldots, \beta_1)$.

The coupled M/G/1 FCFS queue is constructed with these generated random variables

and it has exactly the same unfinished workload as that in the M/G/1 PS model.

2.4 Other perfect sampling algorithms

2.4.1 Regenerative method of perfect sampling

Let $X_n, n = 0, 1, \ldots$ denote the number of customers in a stable queue just before the n^{th} arrival, with $X_0 = 0$. Then $\{X_n : n \geq 0\}$ is a positive recurrent non-delayed discrete-time regenerative process with $x^* = 0$ as the regenerative setting. Denote by $T \in \mathbb{N}$ its cycle length, so $\mathbb{E}(T) < \infty$ [5, p. 9, Theorem 2.2], since the probability of renewal at state 0 is strictly positive. The cycle length is the number of customers served in a busy period. Explicitly, a generic cycle with length T can be defined as

$$C = \{X_n : 0 \leq n < T\}.$$

It is easy to simulate i.i.d. cycles and the sequentially generated ones are denoted by

$$C^{(j)} = \left\{X_n^{(j)} : 0 \leq n < T^{(j)}\right\}, j \geq 1, \tag{2.12}$$

with corresponding cycle lengths $T^{(j)}$.

It is known (see Asmussen and Glynn [6, p. 111]) that

$$\pi_x = \frac{1}{\mathbb{E}(T)}\mathbb{E}\left(\sum_{n=0}^{T-1} \mathbb{1}\{X_n = x\}\right) = \frac{1}{\mathbb{E}(T)}\mathbb{E}\left(\sum_{n=1}^{T} \mathbb{1}\{X_n = x\}\right),$$

where $x = 0, 1, \ldots$.

Denote by T^e the residual length of a randomly selected cycle. Explicitly, suppose X_0 is sampled from the limiting distribution, then

$$T^e = \min\{n \geq 1 : X_n = 0\}, \tag{2.13}$$

It is clear that

$$\Pr(T^e = n) = \frac{\Pr(T \geq n)}{\mathbb{E}(T)}, \text{ where } n = 1, 2, \ldots. \tag{2.14}$$

Let

$$J = \min\{j \geq 1 : T^{(j)} \geq T^e\}, \tag{2.15}$$

then

$$X_{T^e}^{(J)},$$

which is the number of customers in the system found by the $T^e th$ arrival, has the stationary

distribution. See Sigman [52] for the proof.

Proposition 2.4.1 *Denote by* $\{X_n\}_{n\geq 0}$ *and* $\{Y_n\}_{n\geq 0}$, $n = 0, 1, \ldots$, *two stationary queues. They are coupled such that* $X_n \geq Y_n, \forall n \geq 0$, *if* $Y_0 = X_0$. *Let* $C^{(j)}$, T^e *and* J *be as defined in equations* (2.12), (2.13) *and* (2.15) *respectively. We map the r.v.'s generated in* $C^{(J)}$, *so that the dominance coupling is preserved, and we simulate forward from empty state with the mapped r.v.'s to construct* $\{Y_n\}_{n\geq 0}$. *So* Y_{T^e} *is a steady-state draw from the limiting distribution of* $\{Y_n\}_{n\geq 0}$.

Proof As described above, the dominating process $\{X_n\}_{n\geq 0}$ is in steady state at the $T^e th$ step, so the coupled chain is also in steady state at this time. □

Remark If $T^{(J)} = T^e$, then the waiting time is 0, since the $T^{(J)} th$ customer finds an empty system.

To apply this method, we need to find a dominating chain for which we can simulate from its limiting distribution. Assume $\{Y_n\}$ is the stochastic process of interest and $\{X_n\}$ is such a coupled dominating process. Suppose we can sample T^e of the cycle of $\{X_n\}$, and let J be defined as above, then $Y^{(J)}_{T^e}$ is a steady-state draw we need.

Based on the above results, we can describe the general algorithm of perfect sampling of the WCQ, which has common service duration distributions among (possibly) different classes of customer. It proceeds as follows:

(1) Sample T^e of a coupled chain which dominates the WCQ in workload. Set $n = T^e$.

(2) Independently and sequentially simulate the coupled dominator (FCFS) cycles $C^{(j)}$, $j \geq 1$, until we get $T^{(J)}$, where $J = \min\{j \geq 1 : T^{(j)} \geq T^e\}$. In $C^{(J)}$, artificially set the class number of the n^{th} arrival as $\mathbb{K}$, which is the class of interest.

(3) Restore the WCQ with the generated inputs of $C^{(J)}$, output the waiting time of the n^{th} arrival as a steady-state draw for class $\mathbb{K}$ customers' waiting time.

This algorithm is quite appealing, since it does not require the reversibility of the dominating chain. We will see that, in the following chapters, it is much easier to achieve the dominance than reversibility.

However its drawback is (Proposition 2.4.2) that the expected runtime is infinite. In practice, this algorithm might take a very long time to stop.

Proposition 2.4.2 *Let* $T^{(j)}$ *be i.i.d. realizations from the distribution of* T *(i.e. a discrete distribution on* $1, 2, \ldots$*), with* T^e *drawn from a distribution as described above. Let* J *be the smallest value with* $T^{(J)} \geq T^e$. *Then* $\mathbb{E}(J)$ *is finite if and only if* T *has finite support.*

Proof It is clear that

$$\Pr(T^e = n) = \frac{\Pr(T \geq n)}{\mathbb{E}(T)},$$

which is shown in equation (2.14). Given $T^e = n$, J is a geometric random variable with success probability of $\Pr(T \geq n)$. Therefore

$$\mathbb{E}(J|T^e = n) = \frac{1}{\Pr(T \geq n)} = \frac{1}{\Pr(T^e = n)\mathbb{E}(T)}.$$

Unconditionally, we obtain

$$\mathbb{E}(J) = \sum_n \mathbb{E}(J|T^e = n) \Pr(T^e = n) = \sum_n \frac{1}{\mathbb{E}(T)},$$

which is $1/\mathbb{E}(T)$ times the count of n, and the result follows. □

2.4.2 Special method of GI/G/1 FCFS queue perfect sampling

This method was proposed by Ensor and Glynn [14] based on the fact that the stationary waiting times in a GI/G/1 FCFS queue have the same distribution as the maximum value of an underlying random walk. Since an *Exponential Change of Measure* (ECM) (c.f. Asmussen and Glynn [6, p. 129]) is involved, it implies that the service duration should have light tail distribution.

Stationary waiting time in GI/G/1 FCFS queue

Let W_n be the waiting time and B_n the service duration of the n^{th} ($n = 0, 1, 2, \ldots$) customer, and A_n the inter-arrival time between the n^{th} and $(n+1)^{st}$ one. B_n are i.i.d, A_n i.i.d., and they are independent.

Lindley's formula shows that

$$W_{n+1} = \max\{0, W_n + B_n - A_n\}, \tag{2.16}$$

with $W_0 = 0$. Let $X_{n+1} = B_n - A_n$, then we have

$$\begin{aligned} W_1 &= \max\{0, X_1\} \\ W_2 &= \max\{0, W_1 + X_2\} = \max\{0, X_2, X_2 + X_1\} \\ W_3 &= \max\{0, W_2 + X_3\} = \max\{0, X_3, X_3 + X_2, X_3 + X_2 + X_1\} \\ &\dots \\ W_n &= \max\{0, X_n, X_n + X_{n-1}, \dots, X_n + \dots + X_1\}. \end{aligned}$$

Since X_n are i.i.d. , so

$$(X_1, \dots, X_n) \stackrel{D}{=} (X_n, \dots, X_1),$$

i.e. $\{X_n, n \geq 1\}$ is time-reversible. Thus we have

$$W_n \stackrel{D}{=} \max\{0, X_1, X_1 + X_2, \dots, X_1 + \dots + X_n\}.$$

Define $S_n = \sum_{i=1}^{n} X_i$, with $S_0 = 0$, then $\{S_n, n \geq 0\}$ is a random walk. And it yields

$$W_n \stackrel{D}{=} \max_{k=0,\dots,n} \{S_k\},$$

and the stationary waiting time is

$$W_\infty \stackrel{D}{=} \max_{n \geq 0} \{S_n\},$$

which is the maximum value of a random walk. Since this is a stable queue, therefore $\mathbb{E}(A_n) > \mathbb{E}(B_n)$, so $\mathbb{E}(X_n) < 0$. See also Asmussen and Glynn [6, p. 3] or Grimmett and Stirzaker [18, p. 456] for more details.

Algorithm description

(1) Perform ECM.

Let $M_X(t)$ be the m.g.f. and $f_X(x)$ be the p.d.f. of X as noted above. Solve $M_X(\gamma) = 1$ for $\gamma > 0$. Let $\mathbb{P}_\gamma$ be the measure with p.d.f. $e^{\gamma x} f_X(x)$. This is called the ECM.

(2) Construct an increasing process.

Under $\mathbb{P}_\gamma$, it is easy to see that, $\mathbb{E}_\gamma(X) > 0$. Define a strictly increasing process with ladder heights $S_{\tau(n)}$, where

$$\tau(0) = 0, \ \tau(n+1) = \inf\{k > \tau(n) : S_k > S_{\tau(n)}\},$$

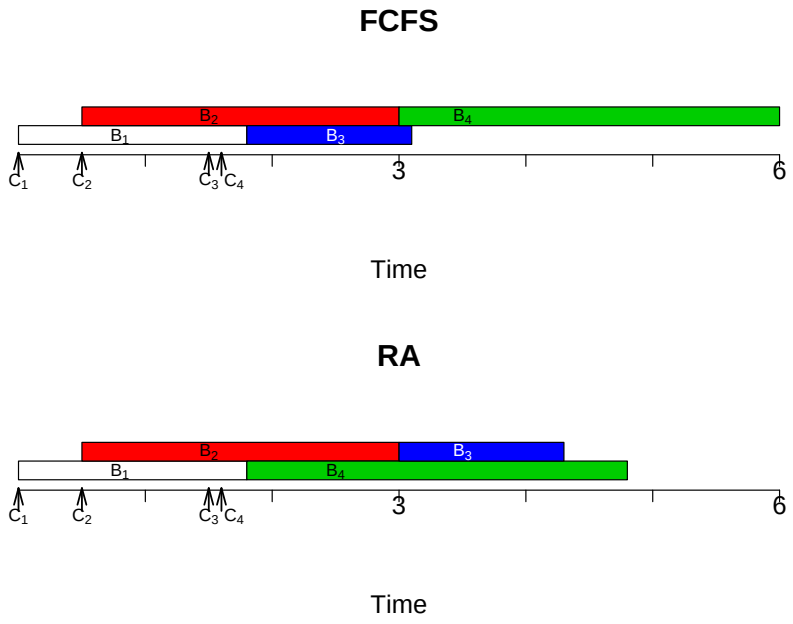

Figure 2.4: RA model is not the sample path upper bound of the simply coupled FCFS queue. There are 4 customers (C_1 through C_4) arriving at times 0, 0.5, 1.5, and 1.6, whose service durations are 1.8, 2.5, 3, and 1.3 respectively. In the RA model, C_1 and C_4 are assigned to one server, and C_2 and C_3 to another. It is obvious that $Q(5) > Q^{RA}(5)$.

(3) Generate W_∞.

Generate $V \sim \text{Exp}(\gamma)$. Let $Z = \sup\{S_{\tau(n)} : S_{\tau(n)} \leq V\}$. Then Z is a stationary draw of W_∞.

For proofs and more details, see Asmussen and Glynn [6, p. 164 and 438].

2.5 Miscellaneous algorithms

2.5.1 Ordinary simulation of GI/G/c FCFS queue with random assignment

This algorithm is introduced because the Random Assignment (RA) model serves as a stochastic upper bound (in unfinished workload) of the coupled FCFS multi-server queue (Wolff [59]). As noted by Wolff [59], it is not the sample path upper bound, which is illustrated in Figure 2.4.

Stochastic upper bound in unfinished workload of multi-server queue

Let $V(t)$ be the unfinished workload at instant t in the FCFS multi-server system, $V^{RA}(t)$ be that in the coupled RA model, and $V^{RO}(t)$ in the coupled RA model with service duration re-ordered according to their arrival order. The "coupled" means these systems are fed with the same arrival instants and associated service durations, and they are both initially empty.

In the RA system, each server has its own queue. A customer is assigned randomly to a sub-queue upon arrival, and the customer has to wait if the designated server is busy, even if there are existing any other free servers. In the RA model, FCFS is violated from the view of whole system, i.e. the order of initiations of service durations could be different to that of the arrivals. But since all customers share the same service duration distribution, by adjusting the order of service, it can be in accordance with that of the arrivals without affecting the distribution of V^{RA}, i.e.

$$V^{RO}(t) =_{so} V^{RA}(t).$$

It is shown (c.f. Asmussen [5, p. 343]) that $V^{RO}(t) \geq V(t)$. So we have the stochastic upper bound as

$$V^{RA}(t) \geq_{so} V(t).$$

With this coupling scheme, it also holds true that

$$Q^{RA}(t) \geq Q(t),$$

where $Q^{RA}(t)$ is the number of customers at time t in the RA model, and $Q(t)$ that in the FCFS queue.

Kiefer-Wolfowitz recursion

As noted by Asmussen [5, p. 341], the ordered unfinished workload at the instant of the n^{th} ($n \geq 0$) arrival is named as Kiefer-Wolfowitz vector. It is denoted as $\vec{W}_n = (W_n^{(1)}, \ldots, W_n^{(c)})$, with $W_n^{(1)} \leq W_n^{(2)} \leq \ldots \leq W_n^{(c)}$.

Let A_n be the inter-arrival time from the n^{th} to the $(n+1)^{st}$, and B_n the service duration of the n^{th} arrival. So

$$\vec{W}_{n+1} = \mathcal{R}\left(\left(W_n^{(1)} + B_n - A_n\right)^+, \left(W_n^{(2)} - A_n\right)^+, \ldots, \left(W_n^{(c)} - A_n\right)^+\right),$$

where $(x)^+ = \max\{0, x\}$, and $\mathcal{R}$ is an operator on $\mathbb{R}^c$ which orders the coordinates in ascending order. This equation is called Kiefer-Wolfowitz recursion. It details the FCFS rule in the way that, the first available server is chosen for the earliest arrival waiting in the queue or to come.

Furthermore, $\{\vec{W}_n\}$ is a Markov chain, and it is used to describe the limiting distribution of the multi-server system.

Simulation algorithm for the RA model

Label servers with 1 through c, and denote $\vec{V}_n = (V_n^{(1)}, \ldots, V_n^{(c)})$ the unfinished workload vector seen by the n^{th} arrival, where $V_n^{(k)}, k = 1, \ldots, c$, is the unfinished workload in server k.

Let l_n be the label number of the server being chosen for the $(n+1)^{st}$ arrival. In the RA model,

$$l_n \sim \text{Unif}\{1, 2, \ldots, c\},$$

and $\vec{V}_{n+1}$ is updated as

$$\begin{cases} V_{n+1}^{(l_n)} = \max\left\{0, V_n^{(l_n)} + B_n - A_n\right\}, \\ V_{n+1}^{(k)} = \max\left\{0, V_n^{(k)} - A_n\right\}, k \in \{1, \ldots, c\}, k \neq l_n. \end{cases}$$

Essentially, this is another form of the Kiefer-Wolfowitz recursion with RA discipline. In terms of computation efficiency, it is preferable.

With this recursive formula, the algorithm of ordinary simulation of the RA model is quite simple:

(1) Initialize $\vec{V}_0 = \mathbf{0}$, which is an all-zero vector,

(2) Repeat the recursive formula with independently generated A_n and B_n until $\vec{V}_N = 0$, found by the N^{th} arrival.

2.5.2 Time-varying Poisson process simulation

Definition

The time-varying (also referred to as "non-homogeneous" or "time-inhomogeneous") process is defined as follows (c.f. Ross [48, p. 78]).

A counting process $\{N(t), t \geq 0\}$ is said to be a non-stationary or non-homogeneous Poisson process with intensity function $\lambda(t), t \geq 0$ if

(i) $N(0) = 0$.

(ii) $\{N(t), t \geq 0\}$ has independent increments.

(iii) $\mathbb{P}\{N(t+h) - N(t) \geq 2\} = o(h)$.

(iv) $\mathbb{P}\{N(t+h) - N(t) = 1\} = \lambda(t)h + o(h)$.

where $\lim_{h\to 0} \frac{o(h)}{h} = 0$.

Let

$$\Lambda(t) = \int_0^t \lambda(s)ds,$$

and let $\Delta\Lambda(t) = \Lambda(t + \Delta t) - \Lambda(t)$, and $\Delta N(t) = N(t + \Delta t) - N(t)$ then it can be shown that

$$\mathbb{P}\{\Delta N(t) = n\} = \frac{[\Delta\Lambda(t)]^n}{n!} e^{-\Delta\Lambda(t)}, n \geq 0.$$

Denote by N_i ($i \in \mathbb{Z}$) the number of events on the consecutive unit intervals $(i-1, i)$. With the periodic assumption for this thesis, it follows that N_i are i.i.d. Poisson random variable with rate $\Lambda(1)$.

The commonly used methods for generating time-varying Poisson processes are listed below (see Ross [49, p. 697-703]).

Simulation: Thinning method

Let $\lambda^* = \sup\{\lambda(t), t \in \mathbb{R}\}$, and assume λ^* exists and is finite. Generate i.i.d. $E_n \sim \text{Exp}(\lambda^*)$, and i.i.d. $U_n \sim \text{Unif}(0, 1)$ ($n \geq 0$). Let t_n be the instants of events of the homogeneous Poisson process, with $t_0 = 0$. The algorithm is quite simple.

Repeat this step until getting desired number of time-varying events:

- Assign $t_{n+1} = t_n + E_n$.

  If

  $$U_{n+1} \leq \frac{\lambda(t_{n+1})}{\lambda^*},$$

  then output t_{n+1} as the instant of the time-varying event.

  Otherwise, output nothing and continue the loop.

Simulation: Order statistics method

This algorithm is used to generate time-varying events on an interval with given length. Without loss of generality, assume the interval be $(0, T)$, and define c.d.f.

$$F(x) = \frac{\Lambda(x)}{\Lambda(T)}, x \in (0, T),$$

which has inverse function as $F^{-1}(y), y \in (0, 1)$.

(1) Generate r.v. $N \sim \text{Poi}(\Lambda(T))$.

(2) Generate N uniform r.v. on $(0, T)$ independently and arrange them in ascending order. I.e.

$$U_{(1)} < \ldots < U_{(N)}.$$

(3) Output $F^{-1}(U_{(k)}), k = 1, \ldots, N$ as the sequential instants of the time-varying events.

For the periodic Poisson process with cycle length 1, the time-varying event instants can be constructed straightforwardly from a coupled homogeneous Poisson process, whose rate is a constant $\Lambda(1)$. Let $t^N \in \mathbb{R}$ and $t^H \in \mathbb{R}$ be the coupled time-varying and homogeneous instants in the same unit interval respectively. Then

$$\begin{aligned} \lfloor t^N \rfloor &= \lfloor t^H \rfloor \\ t^N - \lfloor t^N \rfloor &\sim F(\cdot) \\ t^H - \lfloor t^H \rfloor &\sim \text{Unif}(0, 1), \end{aligned}$$

where

$$F(x) = \frac{\Lambda(x)}{\Lambda(1)}, x \in (0, 1).$$

So

$$t^N = \lfloor t^H \rfloor + F^{-1}(t^H - \lfloor t^H \rfloor) \tag{2.17}$$

Simulation: Inter-event time method

Assume an event occurs at at time x and the time to next event is denoted as T_x. According to the independent increments property, we have the tail probability of T_x as

$$\begin{aligned} \overline{F}_x(t) = \mathbb{P}(T_x > t) &= \mathbb{P}[\text{no events in } (x, x + t)] \\ &= e^{-\int_x^{x+t} \lambda(s)ds} = e^{-\int_0^t \lambda(x+s)ds}. \end{aligned}$$

So

$$F_x(t) = 1 - e^{-\int_0^t \lambda(x+s)ds}. \tag{2.18}$$

Let $t_n (n \geq 0)$ be the instants of the time-varying events, with $t_0 = 0$, then the algorithm can be specified as repetitions of these steps until the desired number of events are achieved.

(1) Simulate T_{t_n} according to the c.d.f. shown in equation (2.18).

(2) Assign $t_{n+1} = t_n + T_{t_n}$.

2.5.3 Gaver-Stehfest inversion of LST

As what the name implies, this method is attributed to two people. Gaver [16] proposed the approximation in a recursive form, and Stehfest [55] refined it by accelerating the convergence.

Assume function $f(x)$ (which could stand for a p.d.f. or c.d.f.) has LST as $\tilde{f}(s)$. Then the approximation of the inversion is

$$f_g(t, M) = \frac{\ln(2)}{t} \sum_{k=1}^{2M} \zeta_k \tilde{f}\left(\frac{k \ln(2)}{t}\right),$$

where

$$\zeta_k = (-1)^{M+k} \sum_{j=\lfloor (k+1)/2 \rfloor}^{\min\{k,M\}} \frac{j^{M+1}}{M!} \left(\begin{array}{c} M \\ j \end{array} \right) \left(\begin{array}{c} 2j \\ j \end{array} \right) \left(\begin{array}{c} j \\ k-j \end{array} \right),$$

$M \in \mathbb{N}$, and $\lfloor x \rfloor$ is the floor function.

This formula was presented in Abate and Whitt [3], with $2M$ corresponding to the number of transform evaluations. Significant digits of this method is around $0.9M$, with requirement for system precision of $2.2M$ digits.

If we use 8-byte floating point numbers, then the system precision is 15. It implies that $\max\{M\} = 6$, which is computed as $\left\lfloor \frac{15}{2.2} \right\rfloor$, and the significant digits of inversion is around 5 ($= \lfloor 0.9 \times 6 \rfloor$).

Chapter 3

Sampling homogeneous queues

In this chapter, we will treat homogeneous WCQs with single or multiple servers. First, we demonstrate an application of CFTP on the GI/G/1 FCFS queue to achieve nearly perfect sampling. Then for the WCQs with non-FCFS disciplines, by taking APQs as examples, we 1) present ordinary simulation methods as elementary tools for perfect sampling algorithms; 2) apply perfect sampling algorithms to the single-server or multi-server queues with common general service distributions; 3) introduce a nearly perfect sampling algorithm, called CFTP Block Absorption, when the service duration distributions vary among different classes of customers.

3.1 Nearly perfect sampling of the GI/G/1 FCFS queue with heavy tail distribution inputs

Before analyzing the WCQs with non-FCFS disciplines, we would like to show an application of CFTP to implement nearly perfect sampling of the GI/G/1 FCFS queue. We know how to simulate the steady-state draw of the GI/G/1 FCFS queues with light tail service duration distributions (see Section 2.4.2), but the heavy tail case is still quite challenging, especially when the inter-arrival time and service duration both have heavy tail distributions.

Let W_t be the unfinished workload at time t. The states just before and after an arrival instant are of interest, denoted by W_{n-} and W_{n+}, where $n \in \mathbb{Z}$. It is clear that W_{n-} is the actual waiting time. As for a transition from W_{n-} to $W_{(n+1)-}$, it can be separated into two stages: 1) the unfinished workload gets a jump due to the service requirement introduced by the arrival; 2) it decreases by 1 per unit time before the next arrival or before the system becomes idle. These

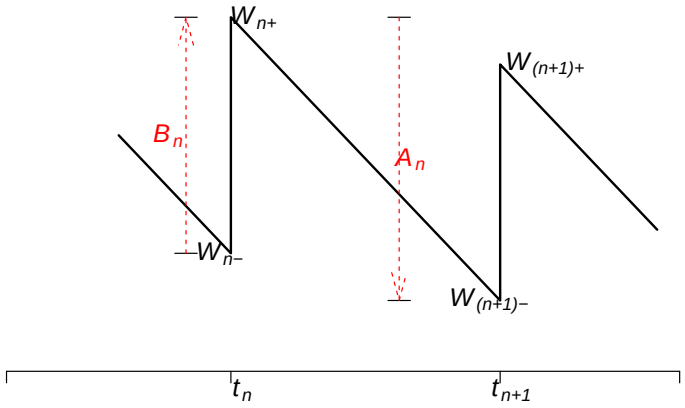

Figure 3.1: One transition of the unfinished workload from W_{n-} to $W_{(n+1)-}$ in the GI/G/1 FCFS queue, where t_n is the arrival instant of the n^{th} customer.

two stages are represented as follows.

$$W_{n+} = W_{n-} + B_n \tag{3.1}$$

$$W_{(n+1)-} = (W_{n+} - A_n)^+, \tag{3.2}$$

where B_n is the service duration of the n^{th} arrival and A_n the inter-arrival time from the n^{th} to the $(n+1)^{st}$ arrival, as illustrated in Figure 3.1.

The Multishift Coupling (Section 2.3.2) will be used in these two stages of transition. In the first stage of transition (equation (3.1)),

$$W_{n+} = E + \left\lceil \frac{W_{n-} - E}{\mathrm{d}} \right\rceil \mathrm{d}, \tag{3.3}$$

where E is simulated from the service duration distribution ($G(\cdot)$) and d is the length of the horizontal line segment intersected with the area under the p.d.f.'s curve (c.f. Figure 2.1) of $G(\cdot)$.

As for the second stage (equation (3.2)),

$$W_{(n+1)-} = \left(-E' + \left\lfloor \frac{W_{n+} + E'}{\mathrm{d}'} \right\rfloor \mathrm{d}' \right)^+, \tag{3.4}$$

where E' is sampled from the inter-arrival time distribution ($\mathcal{A}(\cdot)$) and d' is the length of the horizontal line segment intersected with the area under the p.d.f.'s curve of $\mathcal{A}(\cdot)$. Please see the coming example for details of this coupling method.

Since the Multishift Coupling is monotone, our coupling scheme is also monotone, i.e. $W_{n-} \leq W'_{n-}$ ensures that $W_{(n+1)-} \leq W'_{(n+1)-}$, where the superscript is used to identify the different chain. So we only need to follow the maximum and minimum states of all possible chains.

With regard to the heavy tail queues, the tail probability of the waiting time has this property (see Theorem 9.1 by Asmussen [5, p. 296])

$$\frac{\Pr(W > x)}{\frac{\rho}{1-\rho}\bar{B}_0(x)} \to 1, \text{ as } x \to \infty, \tag{3.5}$$

where W is the steady-state waiting time, $B_0(x)$ stands for the equilibrium distribution of the service duration, and $\bar{B}_0(x) = 1 - B_0(x)$.

Although the waiting time is unbounded, if we start from states (of the waiting time) 0 through x_0 (which is quite large such that $\frac{\rho}{1-\rho}\bar{B}_0(x_0) = \epsilon$), we can come within ϵ of stationary draws (in the total variation sense, see Definition 2.1), if the coalescence happens in the first trial.

3.1.1 An example

We consider a GI/G/1 FCFS queue, whose inter-arrival times and service durations have Pareto (type *II*) distributions. We choose the Pareto distribution because it is often the benchmark indicator of heavy-tailed behaviour. The c.d.f.'s are:

$$\begin{aligned} \mathcal{A}(x) &= 1 - \left(\frac{5}{x+5}\right)^{10}, x > 0, \\ \text{and } G(x) &= 1 - \left(\frac{1}{x+1}\right)^{3}, x > 0, \end{aligned}$$

for the inter-arrival time and service duration respectively. Denote by $f_A(x)$ and $f_B(x)$ the p.d.f.'s of distributions of $\mathcal{A}$ and G respectively. Therefore we have

$$\begin{aligned} f_A(x) &= 2\left(\frac{5}{x+5}\right)^{11}, x > 0, \\ \text{and } f_B(x) &= \frac{3}{(x+1)^4}, x > 0, \end{aligned}$$

It is easy to see that $\rho = 0.9$, $\mathbb{E}(A) = 0.5556$, and $\mathbb{E}(B) = 0.5$, and the equilibrium distribu-

tion of the service duration distribution has c.d.f. as

$$B_0(x) = 1 - \left(\frac{1}{x+1}\right)^2, x > 0,$$

which is also Pareto.

In the first stage of transition, we couple the service durations of all possible chains. As shown in Figure 3.2, there are two curves. One is $y = f_B(x)$, and another is the shifted p.d.f. $y = f_B(x - W_{n-})$. Uniformly draw a point (denoted by C_0) under curve $y = f_B(x)$: sample $E \sim G(\cdot)$ and $U \sim \text{Unif}(0, 1)$, then let

$$D = f_B(E)U. \tag{3.6}$$

Thus (E, D) is the coordinate of point C_0. Draw a horizontal line ($y = D$) across point C_0. It intersects with the area under curve $y = f_B(x)$. The width of the intersected line segment is d, and $\text{d} = H$, which is the x-coordinate of the intersection of curve $y = f_B(x)$ and line $y = D$.

It is clear that $f_B(H) = D$. Combining with equation 3.6, we have

$$\begin{aligned} f_B(E)U = f_B(H) &\Rightarrow H = (E+1)U^{-1/4} - 1 \\ &\Rightarrow \text{d} = (E+1)U^{-1/4} - 1. \end{aligned}$$

Starting from point C_0 draw points $C_i, i = 0, 1, \ldots$, with interval of d. It is easy to see that there is exactly one of such points under each of the shifted p.d.f.'s curve. For any $W_{n-} \in (x_{i-1}, x_i)$, where x_i is the x-coordinate of point C_i and we define $x_{-1} = 0$, the coupled service duration is

$$B_n = x_i - W_{n-}.$$

Therefore $W_{n+} = x_i = E + \left\lceil \frac{W_{n-} - E}{\text{d}} \right\rceil \text{d}$, as shown in equation (3.3).

In the second stage of transition, we couple the inter-arrival times of all possible chains. The procedure is quite similar to what we did above and we have

$$\text{d}' = (E' + 5)U'^{-1/11} - 5,$$

where $E' \sim \mathcal{A}(\cdot)$ and $U' \sim \text{Unif}(0, 1)$.

The differences are those: we turn over the p.d.f. curves horizontally, and the unfinished workload will be truncated to 0 when $W_{(n+1)-}$ is updated to be negative.

For any $W_{n+} \in (x_{i-1}, x_i), i = 1, 2, \ldots$, where x_i is the x-coordinate of point C_i, the coupled

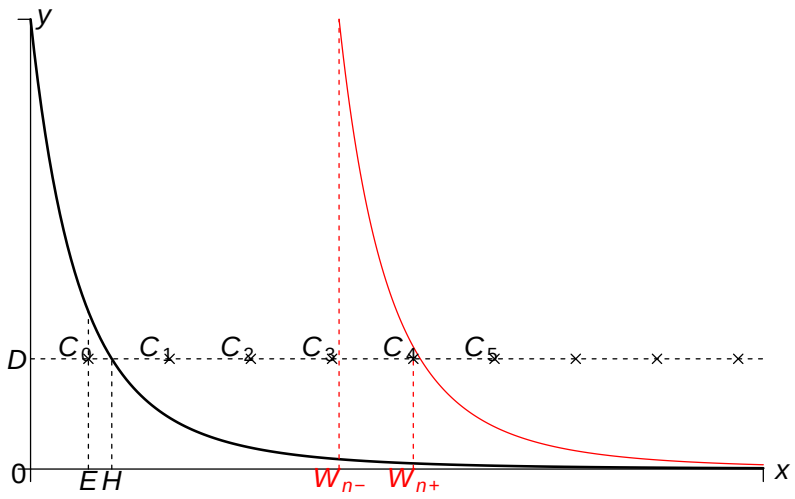

Figure 3.2: Coupling the service durations in the first stage of transition with the Multishift Coupling method.

inter-arrival time is

$$A_n = W_{n+} - x_{i-1}.$$

So $W_{(n+1)-} = (x_{i-1})^+ = \left(-E' + \left\lfloor \frac{W_{n+}+E'}{\mathrm{d}'} \right\rfloor \mathrm{d}'\right)^+$, as demonstrated by equation (3.4). This procedure is illustrated in Figure 3.3.

Because the coupled chains might not coalesce in each trial, we need to consider the bias introduced by ignoring this possibility. Let

$$\begin{aligned} \epsilon_1 &= \Pr(\text{Not capturing the stationary chain with range } (0, x_0) \text{ at } -T_0); \\ \epsilon_2 &= \Pr(\text{No coalescence happens in the first trial}). \end{aligned}$$

According to equation (3.5), we have

$$\epsilon_1 \approx \frac{\rho}{1-\rho} \bar{B}_0(x_0). \tag{3.7}$$

Let $\epsilon = 10^{-10}$ be the total variation distance, then it should be that

$$3\epsilon_1 + \epsilon_2^3 < \epsilon, \tag{3.8}$$

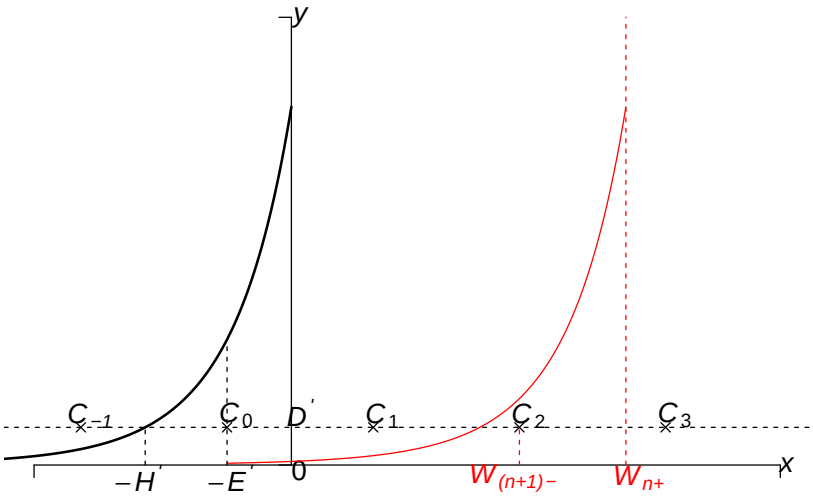

Figure 3.3: Coupling the inter-arrival times in the second stage of transition with the Multishift Coupling method.

by considering a worse case that the coalescence happens in the third trial. Given we have captured the stationary chain at time $-T_m, m = 0, 1$, with range $(0, x_0)$, then the probability of not capturing it at time $-T_{m+1}$ with the same range is less than ϵ_1, because the unfinished workload in this chain is positively correlated. This argument leads to the term of $3\epsilon_1$ in inequality (3.8).

For the second or third trial, the probability of no coalescence occurring is less than ϵ_2. Since these three trials are independent, we obtain the term of ϵ_2^3 in inequality (3.8). The value of ϵ_2 is estimated with the simulated data by using Chebyshev's inequality [49, p. 78]. It is a loose estimation, so if we want to control the probability of bias occurring less than ϵ with one trial, i.e. $\epsilon_1 + \epsilon_2 < \epsilon$, then T_0 will become quite large (with order of 10^{10}). So we consider the multiple trials situation to estimate the upper bound of the bias probability, as illustrated in Figure 3.4.

Since the system evacuates the unfinished workload by $\mathbb{E}(A) - \mathbb{E}(B) = 0.0556$ in each transition, the expected number of transitions for W_{n-} to return the average level is around $x_0/0.0556$). By tuning the value of x_0 and T_0 to accommodate inequality (3.8), we have

$$x_0 = 5.6 \times 10^5 \text{ and } T_0 = 1.41 \times 10^7.$$

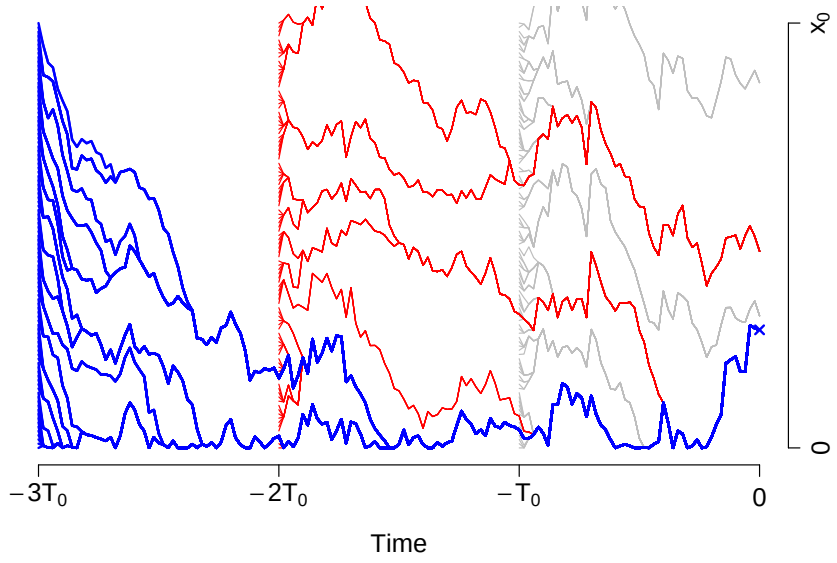

Figure 3.4: A worse case of applying CFTP to the GI/G/1 FCFS queue which has Pareto inter-arrival time and service duration distributions. The coalescence occurs in the third trial.

So with equation (3.7), it follows $\epsilon_1 \approx 2.87 \times 10^{-11}$.

After driving back the extreme large state (x_0) to a normal size with around 10 million steps, the remaining 4 million transitions are enough to ensure these chains with "close" distance to coalesce with high probability. Note that we only need to follow two chains which were started from states 0 and x_0, due to the monotonicity of this coupling.

By repeating the single-trial simulations for 1,000 times, we obtain the numbers of steps till coalescence $X_i, i = 1, \ldots, 1000$. It follows that

$$\begin{aligned} \epsilon_2 &= \Pr(X > T_0) \le \frac{Var(X)}{(T_0 - \mathbb{E}(X))^2} \approx 2.2867 \times 10^{-4} \\ 3\epsilon_1 + \epsilon_2^3 &= 9.81 \times 10^{-11} < \epsilon. \end{aligned}$$

Because the coalescence is a large probability event, all these 1,000 simulations coalesced in the first trial. Table 3.1 demonstrates some statistical results of $X_i, i = 1, \ldots, 1000$ and sampled waiting times. With the sampled waiting times, the e.c.d.f. and its 95% confidence band are plotted in Figure 3.5. For $t > 0$, the standard deviation of $\Pr(W \le t)$ is estimated by $\hat{\sigma} = \frac{\sqrt{\hat{p}(1-\hat{p})}}{\sqrt{N}}$, where $\hat{p} = \frac{\sum_{i=1}^{N} \mathbb{1}(W_i \le t)}{N}$ and $N = 1000$. Therefore the confidence band is approximately $(\hat{p} - 1.96\hat{\sigma}, \hat{p} + 1.96\hat{\sigma})$. Other point-wise confidence bands in this thesis are constructed in the same way.

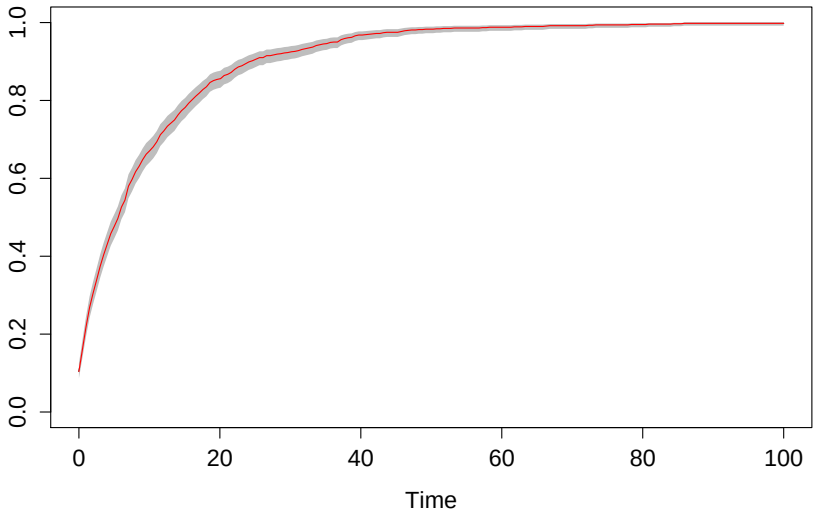

Figure 3.5: The e.c.d.f. from simulations of 1,000 independent draws of waiting times in the GI/G/1 FCFS queue using the CFTP algorithm. Inter-arrival time and service duration both have Pareto distributions. Shaded areas are point-wise 95% confidence bands.

	Transitions for coalescence (X)	**Waiting times**
Min.	9,885,290	0
Max.	10,318,845	335
Average	10,077,061	9.28
Variance	3,700,868,963	250
95% C.I.	(10,073,290, 10,080,831)	(8.29, 10.26)

Table 3.1: Numerical results of the 1,000 simulations of the GI/G/1 FCFS queue

3.2 Ordinary simulation of APQ

Ordinary queueing simulations are implemented by running queues initialized from arbitrarily selected (usually empty) states with specified disciplines for appropriate steps (which is also referred as "burn-in" time) to decrease the impact of the initial states. They are used to verify the analytic solutions or restore the target priority queues in the coupling methods.

3.2.1 $\Sigma^K\mathbf{M}/\mathbf{G}_K/1$ APQ

Denote by C_i ($i \in \mathbb{N}$) the customer who has the i^{th} entry into service. As for C_i, let ξ_i be the service initiation instant, B_i the service duration, τ_i the departure instant, W_i the waiting time and κ_i the class number. It is clear that

$$\tau_i = \xi_i + B_i.$$

Introduce a $K-$length vector $\vec{t_i}$, whose k^{th} ($k = 1, 2, \ldots, K$) entry, $t_i^{(k)}$, denotes the earliest arrival instant of class k customers who have not entered into service by ξ_i+ (just after ξ_i).

So the waiting times of the earliest arrivals at τ_i are

$$\tau_i - t_i^{(k)},\ k = 1, 2, \ldots, K.$$

The system is initialized with $\tau_0 = 0$ and $t_0^{(k)} \sim \text{Exp}(\lambda_k), k = 1, \ldots, K$. For $i = 1, 2, \ldots$, repeat the following.

1. One needs to determine the customer to be selected. There are two possible scenarios as follows.

   - If the server is idle, which means

     $$\max_k \left\{\tau_{i-1} - t_{i-1}^{(k)},\ k = 1, 2, \ldots, K\right\} < 0,$$

     then the earliest arrival to occur is selected for service. So

     $$\kappa_i = \arg\max_k \left\{\tau_{i-1} - t_{i-1}^{(k)},\ k = 1, 2, \ldots, K\right\},$$

     and

     $$\xi_i = t_{i-1}^{(\kappa_i)}.$$

- Otherwise, the server is not idle, and the customer with the maximum accumulated priority is selected, i.e.

$$\kappa_i = \arg\max_k \left\{ b_k \left(\tau_{i-1} - t_{i-1}^{(k)} \right),\ k = 1, 2, \ldots, K \right\},$$

and

$$\xi_i = \tau_{i-1}.$$

In both scenarios, simulate $B_i \sim G_{\kappa_i}(\cdot)$, and update $\tau_i = \xi_i + B_i$.

2. After determining the customer to be selected, its waiting time equals

$$W_i = \xi_i - t_{i-1}^{(\kappa_i)}.$$

3. Update vector $\vec{t_i}$, with

$$\begin{cases} t_i^{(\kappa_i)} = t_{i-1}^{(\kappa_i)} + A^{(\kappa_i)}, \\ t_i^{(k)} = t_{i-1}^{(k)},\ k \neq \kappa_i, \end{cases}$$

where $A^{(\kappa_i)}$ is the inter-arrival time of class κ_i, and $A^{(\kappa_i)} \sim \text{Exp}(\lambda_{\kappa_i})$.

Record the pair of (W_i, κ_i) as the waiting time and corresponding class number of customer C_i.

3.2.2 $\Sigma^K\mathbf{M}/\mathbf{G}_K/c$ APQ

In the multi-server cases, each server is labeled as 1 through c. Upon a new arrival instant, if more than one server is free, the one who became idle earliest is chosen to provide service.

Before customer C_{i+1} is selected for service, the earliest instants of all servers becoming available after accommodating C_i are represented by a c-length vector $\vec{s_i} = \left(s_i^{(1)}, \ldots, s_i^{(c)} \right)$. Let $l_i, i \geq 1$, be the label of the server chosen to provide service to customer C_i.

The system is initialized with $s_0^{(j)} = 0$, $j = 1, \ldots, c$, and $t_0^{(k)} \sim \text{Exp}(\lambda_k), k = 1, \ldots, K$. Repeat recursively as follows for $i = 1, 2, \ldots$.

1. Firstly, choose the server with label

$$l_i = \arg\min_j \left\{ s_{i-1}^{(j)}, j = 1, 2, \ldots, c \right\},$$

taking the smallest index if there is any tie.

2. Then select the customer entering into service and determine the service initiation instant ξ_i.

   If $\max_k \left\{ s_{i-1}^{(l_i)} - t_{i-1}^{(k)},\ k = 1, 2, \ldots, K \right\} < 0$, which means the server being chosen is idle, then

$$\begin{aligned} \kappa_i &= \arg\max_k \left\{ s_{i-1}^{(l_i)} - t_{i-1}^{(k)},\ k = 1, 2, \ldots, K \right\}, \\ \xi_i &= t_{i-1}^{(\kappa_i)}. \end{aligned}$$

   Otherwise

$$\begin{aligned} \kappa_i &= \arg\max_k \left\{ b_k \left[s_{i-1}^{(l_i)} - t_{i-1}^{(k)} \right],\ k = 1, 2, \ldots, K \right\}, \\ \xi_i &= s_{i-1}^{(l_i)}. \end{aligned}$$

   The waiting time of customer C_i is

$$W_i = \xi_i - t_{i-1}^{(\kappa_i)}.$$

3. Update $\vec{s}_i$ and $\vec{t}_i$ as

$$\begin{aligned} s_i^{(l_i)} &= \xi_i + B_i, \text{ and } s_i^{(j)} = s_{i-1}^{(j)}, j \neq l_i, \\ t_i^{(\kappa_i)} &= t_{i-1}^{(\kappa_i)} + A^{(\kappa_i)}, \text{ and } t_i^{(k)} = t_{i-1}^{(k)},\ k \neq \kappa_i, \end{aligned}$$

where $B_i \sim G_{\kappa_i}(\cdot)$ and $A^{(\kappa_i)} \sim \text{Exp}(\lambda_{\kappa_i})$.

Record the pair of (W_i, κ_i) as the waiting time and corresponding class number of customer C_i.

3.3 Perfect sampling of $\Sigma^K \text{M}/\text{G}_K/1$ APQ

In this variant of the APQ, there are K classes of customers, whose arrival processes are Poisson, whereas service duration distributions of the various classes are allowed to differ. We are able to couple it with a FCFS queue by feeding both with the same arrival instants and associated service durations.

The coupled queue is actually an M/G/1 queue under the FCFS discipline. The generic service distribution (B) is a mixture of all classes' ($B^{(i)},\ i = 1, \ldots, K$). As noted in equation

(2.2), c.d.f. of the generic service distribution is

$$G(x) = \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} G_i(x).$$

Since workload in a single-server queue is invariant under all work-conserving disciplines, the sample paths of unfinished workload in the coupled APQ and the FCFS queue are exactly the same. As a result, the latter is used as the dominator.

As presented in Section 2.3.4, the M/G/1 FCFS queue can be simulated backwards, and we know its stationary distribution. Thus the basic idea for perfect sampling of the Σ^KM/G_K/1 APQ is quite straight-forward: generate a stationary state of the dominator at time 0, simulate it backwards until the system become idle, then restore the APQ moving forward in time with inputs generated during the backward simulation. The resulting output, the state at 0, is a steady-state draw of the APQ. Since the service duration distributions differ according to the class numbers, the class numbers need to be generated and their distributions are also different for the customers found at time 0 and those coming afterwards.

3.3.1 Class numbers of the randomly selected service durations

Proposition 3.3.1 *For a randomly selected service duration in the $\Sigma^K M/G_K/1$ PS model, the class number is determined with probability proportional to ρ_i, where $\rho_i = \lambda_i/\mu_i$, $i = 1, \ldots, K$.*

Proof Let $H(x)$ be the c.d.f. of the randomly selected service duration of B, and $H_i(x)$ of $B^{(i)}$.

According to equation (2.10), we have

$$\begin{aligned} H(x) &= \mu \int_0^x \overline{G}(y)dy - \mu x \overline{G}(x), \text{ and} \\ H_i(x) &= \mu_i \int_0^x \overline{G}_i(y)dy - \mu_i x \overline{G}_i(x). \end{aligned}$$

Based on the definition of $G(x)$, it follows that its tail probability has the form of

$$\begin{aligned} \overline{G}(x) &= 1 - \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} G_i(x) = \sum_{i=1}^{K} \left(\frac{\lambda_i}{\lambda} - \frac{\lambda_i}{\lambda} G_i(x) \right) \\ &= \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} (1 - G_i(x)) = \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} \overline{G}_i(x). \end{aligned}$$

So

$$\begin{aligned} H(x) &= \mu \int_0^x \sum_{i=1}^K \frac{\lambda_i}{\lambda} \overline{G}_i(y) dy - \mu x \sum_{i=1}^K \frac{\lambda_i}{\lambda} \overline{G}_i(x) \\ &= \sum_{i=1}^K \left[\int_0^x \frac{\lambda_i}{\lambda/\mu} \overline{G}_i(y) dy - x \frac{\lambda_i}{\lambda/\mu} \overline{G}_i(x) \right] \\ &= \sum_{i=1}^K \frac{\rho_i}{\rho} \left[\int_0^x \mu_i \overline{G}_i(y) dy - x \mu_i \overline{G}_i(x) \right] \\ &= \sum_{i=1}^K \frac{\rho_i}{\rho} H_i(x), \forall x > 0. \end{aligned} \tag{3.9}$$

On the other hand, $H(\cdot)$ is a mixture of $H_i(\cdot), i = 1, \ldots, K$, so

$$H(x) = \sum_{i=1}^K p_i H_i(x), \forall x > 0, \tag{3.10}$$

where p_i is the probability of the randomly selected service duration bearing class number $i, i = 1, \ldots, K$. Because equations (3.9) and (3.10) hold for all $x > 0$, and $H_i(\cdot), i = 1, \ldots, K$, are different distributions (any ties can be merged to one class), it is easy to construct linear equations

$$\sum_{i=1}^K H_i(x_j) p_i = H(x_j), j = 1, \ldots, K,$$

which have a unique solution w.r.t. $(p_1, \ldots, p_K)$ by choosing different values of x_j, $j = 1, \ldots, K$. Equation (3.9) implies that $(\rho_1/\rho, \ldots, \rho_K/\rho)$ is a solution of the linear equations constructed above. So $p_i = \rho_i/\rho, i = 1, \ldots, K$. Thus the result holds. □

3.3.2 Simulating backwards the coupled $\Sigma^K \mathrm{M}/\mathrm{G}_K/1$ FCFS queue

This algorithm follows the framework described in Section 2.3.4. The class numbers of all customers will be identified and recorded. When simulating the PS model forward, at time 0, the class number of a randomly selected service duration is determined with probability proportional to ρ_i. For the following new arrivals, the probability of taking class j is λ_j/λ.

Based on the algorithm description shown in Section 2.3.4, pseudocode of simulating backwards the coupled $\Sigma^K \mathrm{M}/\mathrm{G}_K/1$ FCFS queue is illustrated in Algorithm 1. To make it more readable, some variables are explained in Table 3.2 in the order of occurrence. More details can be found in the algorithm description in Section 2.3.4.

Algorithm 1 Backward simulation of Σ^KM/G$_K$/1 FCFS queue

Initialize INSTANTS, SERVICES and CLASSES to be empty.
Simulate $Q \sim \text{Geom}(1-\rho)$
if $Q = 0$ **then**
  return $\zeta \leftarrow 0$
else
  for $i = 1$ to Q **do**
    Simulate $\mathcal{K}_i$ from $\{1, \ldots, K\}$ with $\Pr(\mathcal{K}_i = k) = \rho_k/\rho$
    Simulate $U_i \sim \text{Unif}(0,1)$; Simulate $X_i \sim H_{\mathcal{K}_i}(\cdot)$
    $Y_i \leftarrow U_i X_i$
  end for
  $t \leftarrow 0$
  Simulate $a \sim \text{Exp}(\lambda)$
  while $Q > 0$ **do**
    $j \leftarrow \arg\min_i\{Y_i, i = 1, \ldots, Q\}$
    $d \leftarrow QY_j$
    if $d < a$ **then**
      $t \leftarrow t + d$; $a \leftarrow a - d$ # departure event
      for $i = 1$ to Q **do**
        $Y_i \leftarrow Y_i - d/Q$
      end for
      Append t to INSTANTS, X_j to SERVICES, and $\mathcal{K}_j$ to CLASSES
      Remove the j^{th} entries in Y, X and $\mathcal{K}$
      $Q \leftarrow Q - 1$
    else
      $t \leftarrow t + a$ # arrival event
      for $i = 1$ to Q **do**
        $Y_i \leftarrow Y_i - a/Q$
      end for
      $Q \leftarrow Q + 1$
      Simulate $\mathcal{K}_Q$ from $\{1, \ldots, K\}$ with $\Pr(\mathcal{K}_Q = k) = \lambda_k/\lambda$
      Simulate $X_Q \sim G_{\mathcal{K}_Q}(\cdot)$
      $Y_Q \leftarrow X_Q$
      Simulate $a \sim \text{Exp}(\lambda)$
    end if
  end while
  $\zeta \leftarrow$ the last element of *Instants*
  Change signs of INSTANTS and reverse orders of INSTANTS, SERVICES and CLASSES
  return -ζ, INSTANTS, SERVICES and CLASSES
end if

Variable	Explanation
Q	Number of customers in the system
ζ	Stationary age of the busy period of M/G/1 FCFS queue
$\mathcal{K}$	Class numbers of customers in the system
X	Service requirements of corresponding customers in the system
Y	Residual service requirements of corresponding customers in the system
t	Event instant
d	Time to next departure event from $t+$
a	Time to next arrival event from $t+$
INSTANTS	Departure instants when running forward, arrival instants when signs changed and order reversed
SERVICES	Corresponding service requirements
CLASSES	Corresponding class numbers

Table 3.2: Variable definitions for Algorithm 1

3.3.3 Restoring the APQ

Based on the algorithm of ordinary simulation of $\Sigma^K \mathrm{M}/\mathrm{G}_K/1$ APQ described in Section 3.2.1, and with inputs provided by Algorithm 1 (ζ, INSTANTS, SERVICES and CLASSES), the stationary draw of the APQ is the state at time 0 of the restored APQ.

Group INSTANTS and SERVICES by class numbers into K sequences of arrival instants: $\left\{t^{(k)}(j)\right\}$, and service durations: $\left\{B^{(k)}(j)\right\}$, $k = 1, \ldots, K$.

Since we are interested in the waiting time of each class of customers, a pseudo-customer of specified class ($\mathbb{K}$) is assumed to arrive at time 0 with corresponding service duration. So the waiting time equals to the instant of its entry into service.

To facilitate computing, generate one more arrival after time 0 for each class with $k \neq \mathbb{K}$, whose arrival instant is simulated from $\mathrm{Exp}(\lambda_k)$, due to the memoryless property of the exponential distribution. Its service duration is simulated from $G_k(\cdot)$.

For class k, append these newly generated arrival instant and service duration to $\left\{t^{(k)}(j)\right\}$ and $\left\{B^{(k)}(j)\right\}$ respectively, where $j = 1, \ldots, N^{(k)}$, and $N^{(k)} \geq 1$.

Introduce a K-length vector $\vec{n}_i = \left(n_i^{(1)}, \ldots, n_i^{(K)}\right)$ to record indices of the earliest arrivals of all classes who have not entered into service by ξ_i+, where ξ_i is the service initiation instant of customer $C_i, i = 1, 2, \ldots$.

It may happen that for some i: $n_i^{(k)} > N^{(k)}$. This means that higher priority customers who arrive after time 0 have "cut in" under the APQ discipline and been served before the pseudo-customer. In such a case additional customers (having arrival instants and associated service durations) will be generated.

The loop terminates when C_{i^*} is the pseudo-arrival, i.e. $n_{i^*-1}^{(\mathbb{K})} = N^{(\mathbb{K})}$ and $\kappa_{i^*} = \mathbb{K}$. We

output ξ_{i^*} as the stationary draw from the waiting time distribution for class $\mathbb{K}$. Based on Proposition 2.3.1 we know that the state at time 0 in the $\Sigma^K M/G_K/1$ APQ is a draw from the stationary distribution. The PASTA (Poisson arrivals see time averages) property allows adding a pseudo-customer at time 0. Since it arrives at a steady state, the statistic related to it (i.e. the waiting time) is also stationary.

Pseudocode of this part is shown in Algorithm 2. In this code, $\omega^{(k)}, k = 1, \ldots, K$, stand for the earliest arrival instants (of customers who is still waiting) found just before the service completion of some customer.

3.3.4 Examples

To validate this sampling based method, we will first compare them with existing analytical results. Theoretical results for the $\Sigma^K M/G_K/1$ classical (absolute) priority queue can be found in Conway et al. [13, p. 163]. Stanford et al. [54] presented the solutions of the waiting time distributions of the $\Sigma^K M/G_K/1$ accumulating priority queue.

To achieve this goal, consider a 2-class priority queue ($\Sigma^2 M/M_2/1$) with parameters

$$(\lambda_1, \lambda_2) = (0.1, 0.5),\ (\mu_1, \mu_2) = (0.2, 1.1). \tag{3.11}$$

For the APQ

$$(b_1, b_2) = (1, 0.5). \tag{3.12}$$

With the dominated CFTP method, we generated 1,000 independent draws from the limiting distribution of the APQ. We have superimposed the plot of the e.c.d.f.'s to compare them with the theoretical ones, which are computed with the Gaver-Stehfest algorithm (see Section 2.5.3). As illustrated in Figure 3.6, they match very well, as the theoretical lines lie well within the 95% confidence interval of estimation.

By changing Line 21 in Algorithm 2 as

$$\kappa \leftarrow \min\left\{i : \tau - \omega^{(i)} \geq 0, i = 1, \ldots, K\right\},$$

the classical priority discipline is implemented. As shown in Figure 3.7, similar comparisons are illustrated between the e.c.d.f.'s and theoretical c.d.f.'s, and the result is also satisfactory.

Remark The priority accumulation function can be defined arbitrarily, e.g. a positive initial value plus the linearly (or quadratically) in time accumulated priority. Our algorithm is indifferent to this setting.

Algorithm 2 Restoring $\Sigma^K\mathrm{M}/\mathrm{G}_K/1$ APQ

if $\zeta = 0$ **then**
  $W \leftarrow 0$
else
  for $i = 1$ to K **do**
    Get sequence $\left\{t^{(i)}(j)\right\}$ and $\left\{B^{(i)}(j)\right\}$ by class number i
    Simulate $X \sim G_i(\cdot)$
    if $i \neq \mathbb{K}$ **then**
      Simulate $E \sim \mathrm{Exp}(\lambda_i)$
    else
      $E \leftarrow 0$
    end if
    Append E to $\left\{t^{(i)}(j)\right\}$ and X to $\left\{B^{(i)}(j)\right\}$
    $n^{(i)} \leftarrow 1$; $N^{(i)} \leftarrow$ the number of records in $\left\{t^{(i)}(j)\right\}$
  end for
  $\tau \leftarrow -\zeta$; $\omega^{(i)} \leftarrow t^{(i)}(n^{(i)}), i = 1, \ldots, K$ # earliest arrival instants
  loop
    if $\max_i \left\{\tau - \omega^{(i)},\ i = 1, .., K\right\} < 0$ **then**
      $\kappa \leftarrow \arg\max_i \left\{\tau - \omega^{(i)},\ i = 1, .., K\right\}$
      $\xi \leftarrow \omega^{(\kappa)}$
    else
      $\kappa \leftarrow \arg\max_i \left\{b_i\left[\tau - \omega^{(i)}\right],\ i = 1, .., K\right\}$
      $\xi \leftarrow \tau$
    end if
    if $\kappa = \mathbb{K}$ and $n^{(\mathbb{K})} = N^{(\mathbb{K})}$ **then**
      $W \leftarrow \xi$; break
    end if
    if $n^{(\kappa)} \leq N^{(\kappa)}$ **then**
      $X \leftarrow B^{(\kappa)}(n^{(\kappa)})$
    else
      Simulate $X \sim G_\kappa(\cdot)$
    end if
    $\tau \leftarrow \xi + X$; $n^{(\kappa)} \leftarrow n^{(\kappa)} + 1$
    if $n^{(\kappa)} \leq N^{(\kappa)}$ **then**
      $\omega^{(\kappa)} \leftarrow t^{(\kappa)}(n^{(\kappa)})$
    else
      Simulate $A \sim \mathrm{Exp}(\lambda_\kappa)$; $\omega^{(\kappa)} \leftarrow \omega^{(\kappa)} + A$
    end if
  end loop
end if
return W

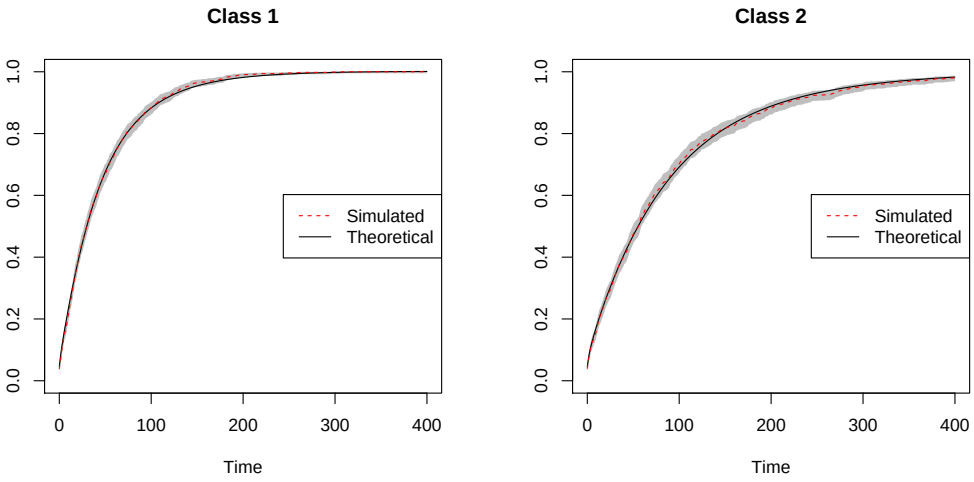

Figure 3.6: The e.c.d.f.'s from simulations of 1,000 independent draws of waiting times in the APQ using the dominated CFTP algorithm, compared with the theoretical c.d.f.'s. Shaded areas are pointwise 95% confidence bands.

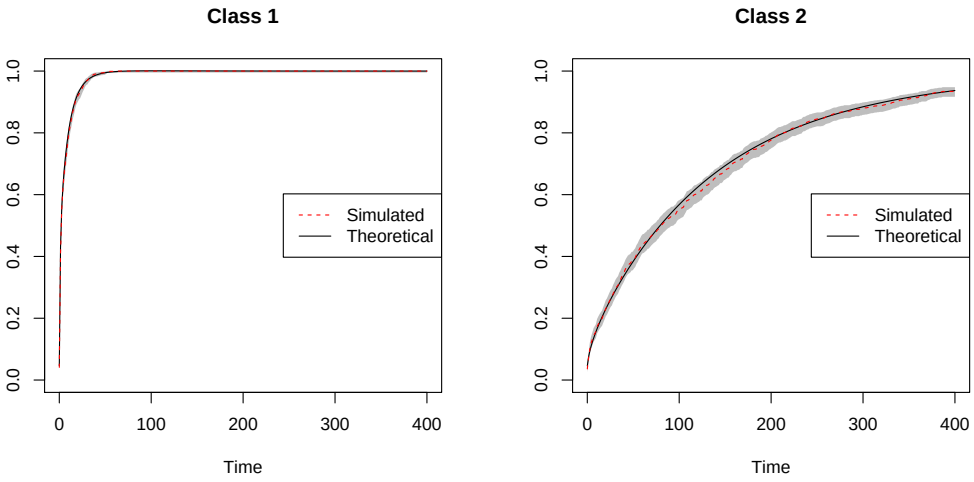

Figure 3.7: The e.c.d.f.'s from simulations of 1,000 independent draws of waiting times in the classical priority queue using the dominated CFTP algorithm, compared with the theoretical c.d.f.'s. Shaded areas are pointwise 95% confidence bands.

3.4 Perfect sampling of Σ^KM/G/c APQ

In this section we consider the case where all classes of customers share a common general distribution.

Sigman [51] presented a perfect sampling algorithm for the M/G/c FCFS queue under a very lightly loaded condition (i.e. $\rho < \frac{1}{c}$). A coupling single-server PS model (which is reversible, see Section 2.3.4) was used as the dominator, then the dominated CFTP followed.

We discuss less restrictive condition $\rho < 1$, and implement perfect sampling with two methods: the regenerative method (RM) and dominated CFTP. Actually, the former is an extended version of that by Sigman [52] with minor changes.

As mentioned before (Section 2.5.1), if the service durations are used in the same order, the RA model dominates the coupled FCFS queue in the numbers of customers in the systems. With the same coupling scheme, it is shown in Proposition 3.4.1 that the same dominance holds between the M/G/c RA model and M/G/c WCQ.

Proposition 3.4.1 *Assume an M/G/c WCQ and an M/G/c RA model are initially (i.e. $t_0 = 0$) empty. They are coupled as follows:*

- *Both are driven by the same arrival instants $\{A_n, n = 1, 2, \ldots\}$ and i.i.d. service durations $\{B_n, n = 1, 2, \ldots\}$.*
- *Let $\{t_n^{WCQ} > t_0, n = 1, 2, \ldots\}$ and $\{t_n^{RA} > t_0, n = 1, 2, \ldots\}$ be the chronological service initiation instants in the M/G/c WCQ and M/G/c RA model respectively. For both queues the customers of the n^{th} chronological service initiation are assigned the same service duration B_n.*

Let Q_t^{WCQ} and Q_t^{RA} be the numbers of customers at time $t > t_0$ in the M/G/c WCQ and M/G/c RA model respectively, then

$$Q_t^{WCQ} \leq Q_t^{RA} \text{ for all } t \geq t_0.$$

Proof We construct a coupled M/G/c FCFS queue with the same arrival instants and service durations as the other two queues. Let $\{t_n^F > t_0, n = 1, 2, \ldots\}$ be its service initiation instants in ascending order, with service duration B_n assigned to the customer entering into service at t_n^F. Let Q_t^F be the number of customers in the FCFS queue.

According to Sigman [52, Lemma 3.1] or Asmussen [5, Lemma 1.3, p. 342], it follows that

$$Q_t^F \leq Q_t^{RA} \text{ for all } t \geq t_0. \tag{3.13}$$

With our construction which ensures the same sequence of service initiations at the same instants, the FCFS rule may differ from the WCQ only in which customer is chosen to enter into service, and not in the instants at which some other customer enters into service. Due to our coupling, the residual workloads of the customers in service and the numbers of customers in these systems will be identical, i.e. $Q_t^F = Q_t^{WCQ}$. Result (3.13) establishes our proposition. □

3.4.1 Using the regenerative method

Based on the algorithm outlines of RM shown in Section 2.4.1, and RA model in Section 2.5.1, the pseudocode is developed as follows.

Firstly, we simulate the stationary excess ($T^e \in \mathbb{N}$) of the cycle of RA model. As shown by Sigman [52], because the arrivals are Poisson processes, according to PASTA (Poisson arrival see time average), at time 0, the stationary unfinished workload in each server can be simulated as in an independent M/G/1 FCFS queue. See Algorithm 3 for details. Please note that if $V = \vec{0}$ at time 0, we still run it forward.

Algorithm 3 Simulation of the stationary excess of M/G/c RA model's cycle

$T^e \leftarrow 0$
for $l = 1$ to c **do**
  $V_l \leftarrow 0$ # V is a vector of unfinished workload
  Simulate $Q \sim \text{Geom}(1-\rho)$
  if $Q > 0$ **then**
    for $j = 1$ to Q **do**
      Simulate $X \sim G_e(\cdot)$ # Equilibrium distribution of service duration
      $V_l \leftarrow V_l + X$
    end for
  end if
end for
repeat
  $T^e \leftarrow T^e + 1$
  Simulate $U \sim \text{Unif}\{1, \ldots, c\}$
  Simulate $B \sim G(\cdot)$
  Simulate $A \sim \text{Exp}(\lambda)$
  $V_U \leftarrow \max\{0, V_U + B - A\}$
  for $l = 1$ to c and $l \neq U$ **do**
    $V_l \leftarrow \max\{0, V_l - A\}$
  end for
until $V = \vec{0}$
return T^e

Then we independently simulate the generic cycles of the RA model, as implemented in

Algorithm 4, which is similar to Algorithm 3, but vector V is initially a vector of zeros. As for the $T^e th$ arrival (the first arrival is denoted as the 0^{th}), we set its class number as $\mathbb{K}$. Note that $T^e \geq 1$ according to the definition (equation 2.13). Let $J = \min\{j \geq 1 : T^{(j)} \geq T^e\}$, then cycle $C^{(J)}$ and its corresponding outcomes (consisting of t^e, INSTANTS, SERVICES, CLASSES and SERVERS) will be used in next step, where t^e is the arrival instant after time 0 of the earliest customer who finds the system is idle. It is used as a sentinel.

Algorithm 4 Simulation of generic busy cycle of the M/G/c RA model

Initialize INSTANTS, SERVICES, CLASSES and SERVERS to be empty.
$t \leftarrow 0$ # Arrival instant
$T \leftarrow 0$ # Length of cycle
$V \leftarrow \vec{0}$ # Unfinished workload vector
repeat
  Simulate $U \sim \text{Unif}(1, \ldots, c)$
  Simulate $B \sim G(\cdot)$
  Simulate $A \sim \text{Exp}(\lambda)$
  if $T = T^e$ **then**
    $\mathcal{K} \leftarrow \mathbb{K}$
  else
    Simulate $\mathcal{K}$ from $\{1, \ldots, K\}$ with $\Pr(\mathcal{K} = k) = \lambda_k/\lambda$
  end if
  Append t to INSTANTS, B to SERVICES, $\mathcal{K}$ to CLASSES and U to SERVERS
  $t \leftarrow t + A$
  $T \leftarrow T + 1$
  $V_U \leftarrow \max\{0, V_U + B - A\}$
  for $l = 1$ to c and $l \neq U$ **do**
    $V_l \leftarrow \max\{0, V_l - A\}$
  end for
until $V = \vec{0}$
return T, $t^e \leftarrow t$, INSTANTS, SERVICES, CLASSES and SERVERS

To apply the coupling scheme specified in Proposition 3.4.1, we need to figure out the service initiation instants of the RA model. Group the arrival instants and corresponding service durations and class numbers by the server labels $(l, l = 1, \ldots, c)$. In server l, apply the FCFS discipline to compute the service initiation instants as INITIATIONS. Assume there are $N \geq 1$ customers in this server. As for customer $C_i, i = 1, \ldots, N$, the arrival instants is t_i, corresponding service duration B_i, service initialization instant ξ_i, and the departure instant τ_i. It is clear that

$$\tau_i = \xi_i + B_i, i = 1, \ldots, N.$$

The algorithm of simulating the initiation instants of the M/G/1 FCFS queue is presented in

Algorithm 5. Aggregate INITIATIONS$_{(l)}$, $l = 1, \ldots, c$, and sort them in ascending order, associated

Algorithm 5 Simulating the initiation instants of the M/G/1 FCFS queue

```
Initialize INITIATIONS to be empty.
τ_0 = 0
for i = 1 to N do
    if τ_{i-1} − t_i ≤ 0 then
        ξ_i = t_i
    else
        ξ_i = τ_{i-1}
    end if
    τ_i = ξ_i + B_i
    Append ξ_i to INITIATIONS
end for
return INITIATIONS
```

with corresponding service durations. Then we have reordered vector of SERVICES′. Arrival instants INSTANTS still keep their order and are associated with corresponding class numbers.

Finally, restore the Σ^KM/G/c APQ with the detailed information to compute the waiting of the $T^e th$ arrival of specified class number $\mathbb{K}$. According to Proposition 2.4.1, the $T^e th$ customer arrives at a steady state, so the related statistic is also stationary.

For writing convenience, service durations (SERVICES′) are recorded in a sequence $\{B_i\}$, and arrival instants are grouped into K sequences: $\left\{t^{(k)}(j)\right\}, k = 1, \ldots, K$ by associated class numbers. Similar to what has been done in restoring the Σ^KM/G$_K$/1 APQ in Algorithm 2, extra customers of class $k, k \neq \mathbb{K}$, are generated to facilitate computation. But the arrival instants are simulated after time t^e, and the $K-1$ service durations are appended to $\{B_i\}$. Let $N^{(k)}$ be the number of arrival instants for class k, and N the number of service durations in $\{B_i\}$, then $N^{(k)} \geq 1$, and $N = \sum_{k=1}^{K} N^{(k)}$. Besides $\vec{n}_i$ and ξ_i (see Algorithm 2), we introduce a c-length vector $\vec{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(c)}\right)$, which is the earliest instants of all servers becoming available after accommodating C_i and before C_{i+1} being selected for service.

Because the pseudo customer of class $\mathbb{K}$ will enter into service in this generic RA cycle, so we need not to simulate customers any more. The pseudocode is available in Algorithm 6.

3.4.2 Using the dominated CFTP

In order to apply the dominated CFTP algorithm to the Σ^KM/G/c APQ, we use the M/G/c RA model as a dominator. Conceptually, we start both models infinitely long ago ($t_0 = -\infty$), coupled as described in Proposition 3.4.1. At time 0, they will both be in steady state. By Proposition 3.4.1, we have $Q_t^{RA} \geq Q_t^A$ for all t, where Q_t^A is the number of customers in the APQ at time t.

Algorithm 6 Restoring Σ^KM/G/c APQ in the Regenerative Method

if $T^e = T$ **then** # T is one of the outputs of Algorithm 4
  $W \leftarrow 0$
else
  for $k = 1$ to K **do**
    Get sequence $\left\{t^{(k)}(j)\right\}$ by class number k
    if $k \neq \mathbb{K}$ **then**
      Simulate $E \sim \text{Exp}(\lambda_k)$; Simulate $X \sim G(\cdot)$
      Append $t^e + E$ to $\left\{t^{(k)}(j)\right\}$ and X to $\{B_i\}$
    end if
  end for
  $N \leftarrow$ the number of elements in $\{B_i\}$
  Initialize $s^{(j)} \leftarrow 0, j = 1, \ldots, c$
  $n^{(k)} \leftarrow 1$; and $\omega^{(k)} \leftarrow t^{(k)}(n^{(k)}), k = 1, \ldots, K$
  for $i = 1$ to N **do**
    $l \leftarrow \arg\min_j \left\{s^{(j)}, j = 1, 2, \ldots, c\right\}$ (taking the smallest index if there is any tie)
    if $\max_k \left\{s^{(l)} - \omega^{(k)},\ k = 1, 2, \ldots, K\right\} < 0$ **then**
      $\mathcal{K} \leftarrow \arg\max_k \left\{s^{(l)} - \omega^{(k)},\ k = 1, .., K\right\}$
      $\xi \leftarrow \omega^{(\mathcal{K})}$
    else
      $\mathcal{K} \leftarrow \arg\max_k \left\{b_k\left[s^{(l)} - \omega^{(k)}\right],\ k = 1, .., K\right\}$
      $\xi \leftarrow s^{(l)}$
    end if
    if $\mathcal{K} = \mathbb{K}$ and $n^{(\mathbb{K})} = N^{(\mathbb{K})}$ **then**
      $W \leftarrow \xi - t^{(\mathbb{K})}(n^{(\mathbb{K})})$; break
    end if
    $s^{(l)} \leftarrow \xi + B_i$
    $n^{(\mathcal{K})} \leftarrow n^{(\mathcal{K})} + 1$
    $\omega^{(\mathcal{K})} \leftarrow t^{(\mathcal{K})}(n^{(\mathcal{K})})$
  end for
end if
return W

The backward simulation of the M/G/c RA model is quite similar to what has been described in Algorithm 2.3.4. Since each class has the same service duration distribution, simulating class numbers can be deferred to the step when the APQ is restored. To satisfy the coupling scheme, the service durations are rearranged according to service initiation instants of the reversed RA model when looking forward in time. Please refer to Algorithm 7 for the pseudocode, where the outcomes of INSTANTS are the arrival instants and SERVICES the rearranged service durations. Notice that the pairing between INSTANTS and SERVICES may differ between the two queues. In order to figure out the initiation instant of each customer, an extra vector named SERVERS is introduced to record the corresponding server labels (denoted by l and $l = 1, \ldots, c$).

Another output of Algorithm 7 is the most recent empty time of the backwards simulated RA system: $-T \in \mathbb{R}$. The coupled Σ^KM/G/c APQ must also be empty at $-T$. We restore the APQ by running forward from the empty state at $-T$, using the outcomes of Algorithm 7, and simulating the class number for each arrival with probability proportional to $\lambda_k, k = 1, \ldots, K$. We output the state at time 0 as the steady-state draw from the Σ^KM/G/c APQ.

Let $\mathbb{K}$ denote the class number whose waiting time is of interest. The procedure to sample a stationary draw from $W_{\mathbb{K}}$ are summarized as below.

1. Simulate the M/G/c RA model backwards as specified in Algorithm 7, and get outputs $-T$, INSTANTS and SERVICES. If $T = 0$ then output $W_{\mathbb{K}} = 0$. Otherwise, continue.

2. Generate a pseudo-arrival of class $\mathbb{K}$ arriving at time 0. Simulate forward from empty state at $-T$ with INSTANTS and SERVICES as described in Section 3.4.3 below. Output the service initiation instant of the pseudo-arrival as a stationary draw from the waiting time distribution for class $\mathbb{K}$ in the Σ^KM/G/c APQ.

Remark According to Proposition 2.3.1, X_0 (which is the state of the Σ^KM/G/c APQ at time 0) is in steady state. Similar argument in Section 3.3.3 supports that the output is a steady-state draw.

3.4.3 Algorithm to restore the Σ^KM/G/c APQ

At time $-T$ Algorithm 7 tells us that the RA queue is empty. By Proposition 3.4.1 we know that the coupled WCQ is also empty. We know the arrival instants of all customers who will arrive before time 0, and a corresponding number of service durations (though we do not yet know the pairing of the customers and service durations). We need to use that information to simulate the WCQ forward in time to find the steady-state draw at time zero. This part of the simulation depends on the particular protocol used; we illustrate in this section using the APQ.

Algorithm 7 M/G/c RA model backward simulation

Initialize vectors INSTANTS, SERVICES and SERVERS to empty.
for $l = 1$ to c **do**
  Simulate $Q_l \sim \text{Geom}(1-\rho)$
  if $Q_l \neq 0$ **then**
    for $i = 1$ to Q_l **do**
      Simulate $U_{l,i} \sim \text{Unif}(0,1)$; Simulate $X_{l,i} \sim H(\cdot)$; $Y_{l,i} \leftarrow U_{l,i}X_{l,i}$
    end for
  end if
end for
$t \leftarrow 0$; Simulate $a \sim \text{Exp}(\lambda)$
if $Q_l = 0$ for all $l = 1, \ldots, c$ **then**
  return $T = 0$
else
  while Exists $Q_l > 0$, $l = 1, \ldots, c$, **do**
    $(l^*, i^*) \leftarrow \arg\min_{l,i}\{Q_l Y_{l,i}, l = 1, \ldots, c; Q_l > 0; i = 1, \ldots, Q_l\}$
    $d \leftarrow Q_{l^*} Y_{l^*,i^*}$
    if $d < a$ **then**
      $t \leftarrow t + d$; $a \leftarrow a - d$ # departure event
      for $l = 1, \ldots, c$ and $i = 1, \ldots, Q_l$, where $Q_l > 0$ **do**
        $Y_{l,i} \leftarrow Y_{l,i} - d/Q_l$
      end for
      Append t to INSTANTS, X_{l^*,i^*} to SERVICES and l^* to SERVERS
      Remove the i^{*th} entries from Y_{l^*} and X_{l^*}
      $Q_{l^*} \leftarrow Q_{l^*} - 1$
    else
      $t \leftarrow t + a$ # arrival event
      for $l = 1, \ldots, c$ and $i = 1, \ldots, Q_l$, where $Q_l > 0$ **do**
        $Y_{l,i} \leftarrow Y_{l,i} - a/Q_l$
      end for
      Simulate $l' \sim \text{Unif}\{1, \ldots, c\}$ # randomly choose a server
      $Q_{l'} \leftarrow Q_{l'} + 1$; Simulate $X_{l',Q_{l'}} \sim G(\cdot)$; $Y_{l',Q_{l'}} \leftarrow X_{l',Q_{l'}}$
      Simulate $a \sim \text{Exp}(\lambda)$
    end if
  end while
  $T \leftarrow$ the last element in INSTANTS
  Change signs of INSTANTS
  Reverse orders of INSTANTS, SERVICES and SERVERS
  Group INSTANTS and SERVICES by server labels as INSTANTS$_l$ and SERVICES$_l$, $l = 1, \ldots, c$.
  Apply the FCFS discipline to INSTANTS$_l$ and SERVICES$_l$, then get INITIATIONS$_l$, $l = 1, \ldots, c$.
  Merge INITIATIONS$_l$ and SERVICES$_l$ back into INITIATIONS and SERVICES. Sort the pairs in ascending order of INITIATIONS.
  return $-T$, INSTANTS and SERVICES.
end if

First we allocate the customers to the various priority classes independently in proportion to their arrival rates, i.e. assign class k with probability λ_k/λ. Break the arrivals up by class into K sequences of arrival instants: $\{t^{(k)}(j)\}$, where ascending orders are preserved for all classes. Let $\{B_j\}$ stand for SERVICES.

In addition to the pseudo-arrival of class $\mathbb{K}$ at time 0, to facilitate computing, generate one more arrival after time 0 for each class with $k \neq \mathbb{K}$, whose arrival instant is simulated from Exp(λ_k), due to the memoryless property of the exponential distribution. Independently simulate K service durations, governed by $G(\cdot)$, and append them to $\{B_j\}$.

Denote by $N^{(k)}$ the number of arrivals of class k, and note that $N^{(k)} \geq 1$. So the K arrival sequences become $\{t^{(k)}(j), j = 1, \ldots, N^{(k)}\}$, $k = 1, \ldots, K$, and the service sequence $\{B_j, j = 1, \ldots, N\}$, where $N = \sum_{k=1}^{K} N^{(k)}$.

Denote by $C_i, i = 1, 2, \ldots$, the customer who is the i^{th} to enter service, by $l_i \in \{1, \ldots, c\}$ the corresponding label of the server chosen to provide service, by κ_i the class number and by ξ_i the instant of service initiation. The values of l_i, κ_i and ξ_i will be determined in the loop described below.

Introduce a K-length vector $\vec{n}_i = \left(n_i^{(1)}, \ldots, n_i^{(K)}\right)$ to record indices of the earliest arrivals of all classes who have not entered into service by ξ_i+. By construction $\vec{n}_i$ exists for all $\xi_i \leq 0$.

Prior to C_{i+1} being selected for service, the earliest instants of all servers becoming available after accommodating C_i are represented by a c-length vector $\vec{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(c)}\right)$.

The system is initialized at $-T$ with $s_0^{(j)} = -T$, $j = 1, \ldots, c$ and $\vec{n}_0 = (1, \ldots, 1)$.

Repeat the following loop for $i = 1, 2, \ldots$.

1. Choose the server with label

$$l_i = \arg\min_j \left\{s_{i-1}^{(j)}, j = 1, 2, \ldots, c\right\},$$

taking the smallest index in the event of a tie.

2. Select the customer entering into service and determine ξ_i.

If $\max_k \left\{s_{i-1}^{(l_i)} - t^{(k)}\left(n_{i-1}^{(k)}\right),\ k = 1, 2, \ldots, K\right\} < 0$, i.e. the server being chosen is idle, then choose the next arrival, i.e.

$$\begin{aligned} \kappa_i &= \arg\max_k \left\{s_{i-1}^{(l_i)} - t^{(k)}\left(n_{i-1}^{(k)}\right),\ k = 1, 2, \ldots, K\right\}, \\ \xi_i &= t^{(\kappa_i)}\left(n_{i-1}^{(\kappa_i)}\right). \end{aligned}$$

Otherwise choose the customer with the highest accumulated priority at this instant, i.e.

$$\begin{aligned}\kappa_i &= \arg\max_k \left\{ b_k \left[s_{i-1}^{(l_i)} - t^{(k)}\left(n_{i-1}^{(k)}\right)\right],\ k = 1, 2, \ldots, K \right\}, \\ \xi_i &= s_{i-1}^{(l_i)}.\end{aligned}$$

3. Update $\vec{s}_i$ and $\vec{n}_i$ as

$$\begin{aligned}s_i^{(l_i)} &= \xi_i + B_i, \text{ and } s_i^{(j)} = s_{i-1}^{(j)}, j \neq l_i, \\ n_i^{(\kappa_i)} &= n_{i-1}^{(\kappa_i)} + 1, \text{ and } n_i^{(k)} = n_{i-1}^{(k)}, k \neq \kappa_i.\end{aligned}$$

It may happen for some i that $n_i^{(k)} > N^{(k)}$. This means that higher priority customers who arrive after time 0 have "cut in" under the APQ discipline and been served before the pseudo-arrival. In such a case additional arrivals and new service durations will be generated. The loop terminates when C_{i^*} is the pseudo-arrival, i.e. $n_{i^*-1}^{(\mathbb{K})} = N^{(\mathbb{K})}$ and $\kappa_{i^*} = \mathbb{K}$. We output ξ_{i^*} as the stationary draw from the waiting time distribution for class $\mathbb{K}$. Corresponding pseudocode can be found in Algorithm 8.

Algorithm 8 Restoring Σ^KM/G/c APQ in the dominated CFTP

if $T = 0$ **then**
  $W \leftarrow 0$
else
  for $k = 1$ to K **do**
    Get sequence $\left\{t^{(k)}(j)\right\}$ by class number k
    if $k \neq \mathbb{K}$ **then**
      Simulate $E \sim \text{Exp}(\lambda_k)$
    else
      $E \leftarrow 0$
    end if
    Simulate $X \sim G(\cdot)$
    Append E to $\left\{t^{(k)}(j)\right\}$ and X to $\left\{B_j\right\}$
  end for
  $N \leftarrow$ the number of elements in $\left\{B_j\right\}$
  $N^{(k)} \leftarrow$ the number of elements in $\left\{t^{(k)}(j)\right\}, k = 1, \ldots, K$
  Initialize $s^{(j)} \leftarrow 0, j = 1, \ldots, c$
  $n^{(k)} \leftarrow 1; \quad \omega^{(k)} \leftarrow t^{(k)}(n^{(k)}), k = 1, \ldots, K;$
  $m \leftarrow 1$ # counter of the service durations being used
  loop
    $l \leftarrow \arg\min_j \left\{s^{(j)}, j = 1, 2, \ldots, c\right\}$ # taking the smallest index if there is any tie
    if $\max_k \left\{s^{(l)} - \omega^{(k)},\ k = 1, 2, \ldots, K\right\} < 0$ **then**
      $\mathcal{K} \leftarrow \arg\max_k \left\{s^{(l)} - \omega^{(k)},\ k = 1, .., K\right\}$
      $\xi \leftarrow \omega^{(\mathcal{K})}$
    else
      $\mathcal{K} \leftarrow \arg\max_k \left\{b_k\left[s^{(l)} - \omega^{(k)}\right],\ k = 1, .., K\right\}$
      $\xi \leftarrow s^{(l)}$
    end if
    if $\mathcal{K} = \mathbb{K}$ and $n^{(\mathbb{K})} = N^{(\mathbb{K})}$ **then**
      $W \leftarrow \xi; \quad$ break
    end if
    if $m \leq N$ **then**
      $X \leftarrow B_m$
    else
      Simulate $X \sim G(\cdot)$
    end if
    $s^{(l)} \leftarrow \xi + X; \quad m \leftarrow m + 1; \quad n^{(\mathcal{K})} \leftarrow n^{(\mathcal{K})} + 1$
    if $n^{(\mathcal{K})} \leq N^{(\mathcal{K})}$ **then**
      $\omega^{(\mathcal{K})} \leftarrow t^{(\mathcal{K})}(n^{(\mathcal{K})})$
    else
      Simulate $A \sim \text{Exp}(\lambda_{\mathcal{K}}); \quad \omega^{(\mathcal{K})} \leftarrow \omega^{(\mathcal{K})} + A$
    end if
  end loop
end if
return W

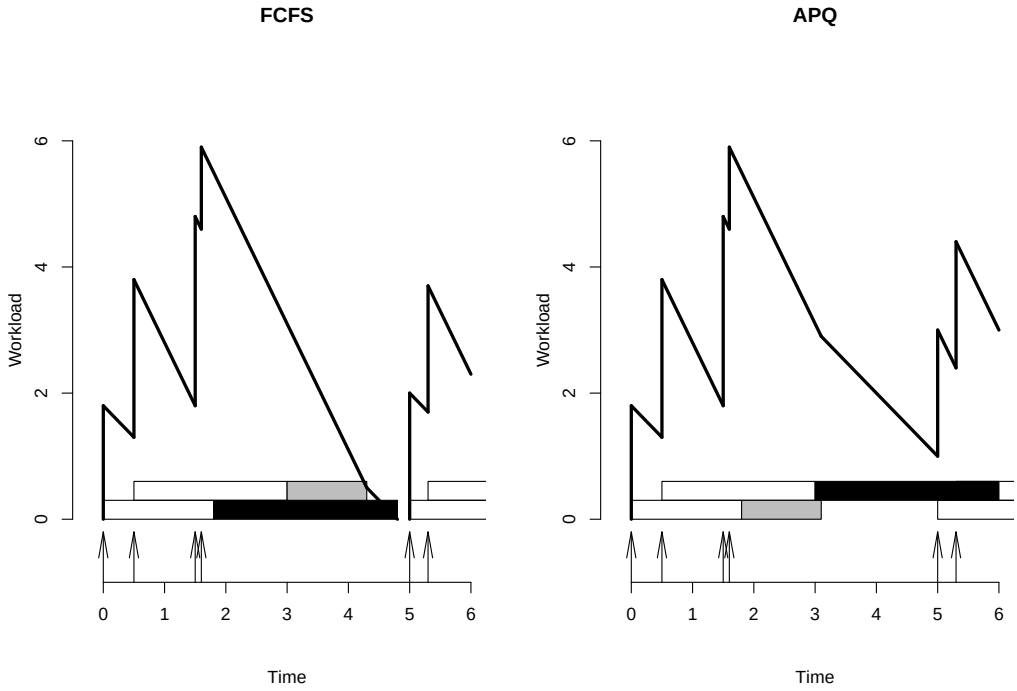

Figure 3.8: Totally idle periods of the coupled $M/G/c$ FCFS and $\Sigma^K M/G_K/c$ APQ, which do not match in this sample realization. The horizontal bars are service durations associated with customers. The gray bar corresponds to a class 1 customer and the black bar a class 2 customer. The upward arrows indicate the arrival instants. The class 2 customer arrives at time 1.5, and the class 1 customer at 1.6. The first departure occurs at 1.8. Under the FCFS discipline, the first busy cycle ends at $\tau_1 = 5$. After applying the APQ discipline ($b_1 = 1, b_2 = 0.5$), the workload excess (unfinished workload at τ_1) interferes the second busy cycle.

3.5 Nearly Perfect sampling of $\Sigma^K M/G_K/c$ APQ

When the service duration distributions for various classes of customers differ, the order in which they enter into service can affect the distribution of the totally idle period. In fact the totally idle periods of the FCFS queue and APQ might not match. As illustrated in Figure 3.8, the unfinished workload at the end of the first busy cycle of the FCFS queue is no longer 0 after applying the APQ discipline. We call the APQ remaining workload the **"workload excess"** from previous busy cycles. It is easy to see that the paths of the $M/G_K/c$ RA model also do not dominate those of the coupled $\Sigma^K M/G_K/c$ APQ under our coupling; this means the Section 3.4 algorithm will not work in this model. We still share arrival instants, but now we assign the service duration to the customer upon arrival. We do not achieve perfect sampling, but using the algorithm described below we can come within ϵ of the true limit (in the total variation sense) for any pre-specified ϵ.

Let $-T_0 \in \mathbb{R}$ be the most recent time in the past when the FCFS queue was totally idle. The coupled $\Sigma^K M/G_K/c$ APQ might not be idle then, if its previous busy period extended across the totally idle period of the $M/G/c$ FCFS queue, so we can't assume that our coupled queue is idle. Since we don't start from an empty system, we can't predict exactly what the state will be at time zero. Fortunately, the potential workload excess is not large (it is likely to be absorbed by subsequent totally idle periods), and in fact, we can compute probabilistic bounds on its size. We will construct our coupling so that the probability of the sample being affected is less than ϵ; if it is not affected, the usual CFTP argument shows that our draw is from the limiting distribution.

This result can be explicitly presented as below by extending Proposition 2.3.1.

Proposition 3.5.1 *Suppose two queues, denoted by $\{X_t\}_{t\in\mathbb{R}}$ and $\{Y_t\}_{t\in\mathbb{R}}$, are stable and they are coupled such that the upper bound of X_t can be determined with Y_t at some time points. $\{Y_t\}_{t\in\mathbb{R}}$ is stationary in time and can be simulated backwards. Assume at a past time $-T$, $X_{-T} \le Y_{-T}+\Delta$, where $T > 0$, $T \in \mathbb{R}$ and $0 \le \Delta < \infty$. We run forward $\{X_t\}_{t\ge -T}$ with $X_{-T} = Y_{-T}$ and the r.v.'s (denoted by $\{U_i, i \in \mathbb{N}\}$ which consists of arrival and service events) generated in the backward simulation of $\{Y_t\}_{-T\le t\le 0}$. If Δ can be absorbed completely by the extra work capacity (e.g. the idle periods) in $\{X_t\}_{-T\le t\le 0}$, then X_0 is a steady-state draw from the limiting distribution of $\{X_t\}_{t\in\mathbb{R}}$.*

Proof Assume these two queues are run from infinitely long ago, so at time 0 both are in steady state. Denote by $\{X_t^x\}_{t\in\mathbb{R}}$ all possible chains with values of x at time $-T$, where $x \in [0, Y_{-T}+\Delta]$. They are driven by a common sequence of r.v.'s $\{U_i, i \in \mathbb{N}\}$. It is easy to verify this coupling is monotone. So if there is empty instant in $\{X_t^{Y_{-T}+\Delta}\}_{-T\le t<0}$, i.e.$\exists t \in [-T, 0), \ni X_t^{Y_{-T}+\Delta} = 0$, then these chains coalesce by time 0, because $\{X_t^{Y_{-T}+\Delta}\}_{-T\le t<0}$ is the upper bound and 0 the intrinsic lower bound. So the unique-valued $X_0 = X_0^{Y_{-T}+\Delta}$ must be the steady-state draw since the chain has been run from infinitely long ago.

Because of the coalescence, $X_0^{Y_{-T}} = X_0^{Y_{-T}+\Delta}$. Therefore we can start with $X_{-T} = Y_{-T}$ and $\{U_i, i \in \mathbb{N}\}$ to construct $\{X_t^{Y_{-T}}\}_{-T\le t\le 0}$, and output $X_0^{Y_{-T}}$ as the steady-state draw. □

Remark (1) We allow $\{X_t\}_{t\in\mathbb{R}}$ to be non-stationary in time, e.g. the time-varying queue. In this case, we get a steady-state draw from the time dependent limiting distribution. So this claim also works in Chapter 4.

(2) Unlike most CFTP implementations, our algorithm allows direct calculation of the (random) starting time in the past, rather than the usual trial and error approach.

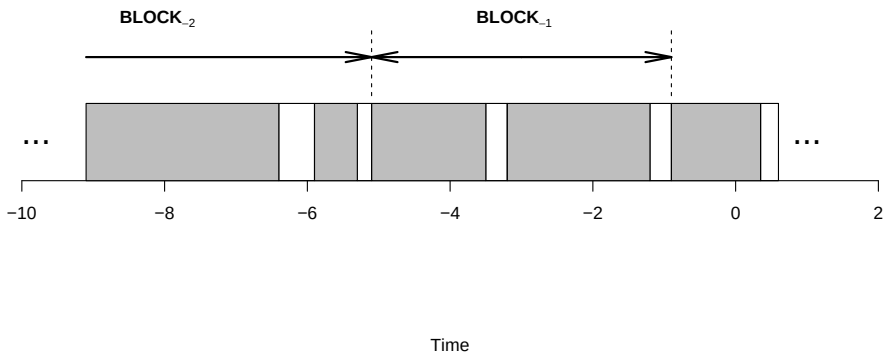

Figure 3.9: Blocks in the M/G/c FCFS queue. The gray rectangles are busy periods and blank ones are totally idle periods. A gray rectangle and a following blank one form a busy cycle. In this case, there are two busy cycles in block $\mathbb{C}_{-1}$ and two ore more busy cycles in $\mathbb{C}_{-2}$.

3.5.1 Workload excess and carryover probability

In the coupled FCFS queue (which can be still treated as an M/G/c FCFS queue with common mixture service duration distribution, see equation (2.2)), we group m_i ($i \in \mathbb{Z}$) consecutive busy cycles together as block $\mathbb{C}_i$, with $\mathbb{C}_{-1}$ being the most recent block before time 0 such that there are m_{-1} totally idle periods between the beginning of $\mathbb{C}_{-1}$ and 0 (note: if one of these periods contains time 0, it is still counted). We will specify m_i below. Assume block $\mathbb{C}_i$ ends at time τ_i with the first arrival starting block $\mathbb{C}_{i+1}$, so $\tau_{-1} < 0 < \tau_0$ if the system is not empty at time 0. Please refer to Figure 3.9 for an illustration.

We now consider the coupled APQ. We call the workload excess generated within $\mathbb{C}_i$ at τ_i the **individual workload excess** and it is denoted by $\Omega_i \geq 0$.

Notice that in $\mathbb{C}_i$ there are m_i totally idle periods, which could absorb the workload excess from previous blocks. Let $\mathcal{G}_i$ denote the summation of the durations of these totally idle periods, then

$$\mathcal{G}_i \sim \text{Gamma}(m_i, \lambda), \tag{3.14}$$

where m_i is the shape and $\lambda = \sum_{k=1}^{K} \lambda_k$ the rate parameter of the Gamma distribution.

The basic idea of this algorithm is that, with the inputs generated in the coupled FCFS queue, we construct the coupled APQ starting from the empty state at time τ_{-2} (i.e. the beginning of block $\mathbb{C}_{-1}$), run it forward and output the state at time 0 as if it was a steady-state draw.

Let $\mathcal{E}$ be the event that $\Omega_{-2} > \mathcal{G}_{-1}$ with $m_{-2} = \infty$, i.e. there might be **carryover** of the workload excess traversing through the interval $(\tau_{-2}, 0)$ after applying the APQ discipline. When $\mathcal{E}$ occurs the simulated workload at time 0 would possibly underestimate the stationary value, whereas in the complementary case it definitely matches it.

For finite m_{-j} ($j = 1, 2, \ldots$), we let $\mathcal{E}_{-j}$ be the event $\Omega_{-(j+1)} > \mathcal{G}_{-j}$, i.e. the individual workload excess from block $\mathbb{C}_{-(j+1)}$ exceeds the summed time fractions of totally idle periods in $\mathbb{C}_{-j}$. If none of $\mathcal{E}_{-j}$, $j = 1, 2, \ldots$, occur then there is no carryover. So we have

$$\bigcap_{j=1}^{\infty} \mathcal{E}_{-j}^c \subset \mathcal{E}^c \Rightarrow \left(\bigcup_{j=1}^{\infty} \mathcal{E}_{-j} \right) \supset \mathcal{E}$$

$$\begin{aligned} \Rightarrow \Pr(\mathcal{E}) &\leq \Pr\left(\bigcup_{j=1}^{\infty} \mathcal{E}_{-j} \right) \\ &\leq \sum_{j=1}^{\infty} \Pr(\mathcal{E}_{-j}) \\ &= \sum_{j=1}^{\infty} \Pr\left[\Omega_{-(j+1)} > \mathcal{G}_{-j} \right]. \end{aligned} \tag{3.15}$$

In the following sections we will establish that items of the summation in equation (3.15) decay in a geometric form, so the sum is finite. We will choose m_{-j} to bound this sum.

3.5.2 Upper bound of individual workload excess

Proposition 3.5.2 *The individual workload excess of $\mathbb{C}_i$ after applying the APQ rule is less than $(c-1)$ times the maximum service duration ($\mathbb{B}_i^*$) in $\mathbb{C}_i$:*

$$\Omega_i < (c-1)\mathbb{B}_i^*. \tag{3.16}$$

Proof At time t, let V_t be the unfinished workload function, with V_t^f and V_t^a being those in the coupled FCFS and APQ respectively. Then

$$\Omega_i = V_{\tau_i}^a.$$

Define the slope of V_t at time t as

$$\beta(t) = \begin{cases} dV_t/dt, & \text{at differentiable points,} \\ \min\{\beta(t-), \beta(t+)\}, & \text{at non-differentiable points.} \end{cases}$$

It is clear that $\beta(t) \in \{-c, -(c-1), \ldots, -1, 0\}$. Similarly, $\beta^f(t)$ and $\beta^a(t)$ correspond to the FCFS and APQ respectively.

After applying the APQ rule, there are two possible cases at time τ_i:

(1) $\beta^a(\tau_i) > -c$, which means there is at least one server idle at this time, and there are at most $(c-1)$ jobs remaining in process, thus inequality (3.16).

(2) $\beta^a(\tau_i) = -c$, i.e. all servers are busy at τ_i. Since it cannot be that all servers keep simultaneously busy throughout (τ_{i-1}, τ_i) (as the total work done will match that of the FCFS queue), so $\exists\, t^* \in (\tau_{i-1}, \tau_i) : \beta^a(t^*-) = -(c-1)$, and $\beta^a(t) = -c, \forall\, t \in (t^*, \tau_i)$, i.e. there are exactly $(c-1)$ unfinished jobs at t^*-. So

$$V^a_{t^*-} < (c-1)\mathbb{B}^*_i.$$

Assume there are N' arrivals on (t^*-, τ_i), and their service durations are $B'_l, l = 1, \ldots, N'$. According to the definition of $\mathbb{C}_i$, it is clear that $V^f_{\tau_i} = 0$, as this is the end of the block. Therefore

$$\begin{aligned}
V^a_{\tau_i} &= V^a_{t^*-} + \sum_{l=1}^{N'} B'_l + (-c)(\tau_i - t^*), \\
&< (c-1)\mathbb{B}^*_i + \sum_{l=1}^{N'} B'_l + \int_{t^*}^{\tau_i} \beta^f(t)dt, \\
&= (c-1)\mathbb{B}^*_i - V^f_{t^*-} + V^f_{t^*-} + \sum_{l=1}^{N'} B'_l + \int_{t^*}^{\tau_i} \beta^f(t)dt, \\
&= (c-1)\mathbb{B}^*_i - V^f_{t^*-} + V^f_{\tau_i} \\
&= (c-1)\mathbb{B}^*_i - V^f_{t^*-} \\
&\leq (c-1)\mathbb{B}^*_i.
\end{aligned}$$

This establishes the result. □

Remark: Proposition 3.5.2 also holds true for other WCQs.

Since the maximum service duration in a block relies on the maxima of the busy cycles in it, we would like to analyze the latter further.

Proposition 3.5.3 *In a busy period of an M/G/c FCFS queue, denote the service durations chronologically according to their arrival instants by $B_0, B_1, \ldots, B_{N-1}$, where N is the number of customers served in this busy period. Let*

$$B^* = \max_k \{B_k, k = 0, \ldots, N-1\}.$$

Then

$$\Pr(B^* > x) \leq \mathbb{E}(N)\overline{G}(x),$$

where $\overline{G}(x)$ is the tail probability of the generic service duration.

Proof Boxma [10] stated a result equivalent to this proposition. We use renewal theory to give an explicit proof.

If the distributions of the B_i's did not depend on N, the result would obviously follow, but the distribution of the B_i's will generally depend on N. Denote by $t_i, i = 0, 1, \ldots$, the chronological arrival instants of customers, which have corresponding service durations $B_0, B_1, \ldots$.

Construct a regenerative process $\{X_t\}_{t\in\mathbb{T}}$, where $\mathbb{T} = [0, \infty)$, as:

$$X_t = B_i,\ t \in [t_i, t_{i+1}),$$

and let $X_i = X_{t_i}$. The embedded renewal process is the sequence of initial instants of the busy cycles.

According to Asmussen [5, Corollary 1.4, p. 171], we have

$$\mathbb{E}_e(f(X_i)) = \frac{1}{\mathbb{E}(N)}\mathbb{E}\sum_{k=0}^{N-1} f(X_k) \Leftrightarrow \mathbb{E}_e(f(B_i)) = \frac{1}{\mathbb{E}(N)}\mathbb{E}\sum_{k=0}^{N-1} f(B_k)$$

for any measurable $f(\cdot)$ where $\mathbb{E}_e$ corresponds to the stationary (or marginal) measure.

Let $f(y) = \mathbb{1}(y > x)$ be the standard indicator function of an event, then it follows that

$$\mathbb{E}_e(\mathbb{1}(B_i > x)) = \frac{1}{\mathbb{E}(N)}\mathbb{E}\sum_{k=0}^{N-1} \mathbb{1}(B_k > x). \tag{3.17}$$

It is clear that

$$\begin{aligned}
&\sum_{k=0}^{N-1} \mathbb{1}(B_k > x) \geq \mathbb{1}(B^* > x) \\
&\Rightarrow \mathbb{E}\sum_{k=0}^{N-1} \mathbb{1}(B_k > x) \geq \mathbb{E}\mathbb{1}(B^* > x) = \Pr(B^* > x).
\end{aligned}$$

Notice that the LHS of equation (3.17) is $\overline{G}(x)$. So we have

$$\overline{G}(x) \geq \frac{1}{\mathbb{E}(N)} \Pr(B^* > x),$$

which is the result desired. □

Proposition 3.5.4 *Let N be the number of customers served in an $M/G/c$ FCFS busy period, then*

$$\mathbb{E}(N) \leq (1-\rho)^{-c}.$$

Proof It is well known (e.g. Wolff [59] and Asmussen [5, p. 342]) that the $M/G/c$ RA model is a stochastic upper bound of the $M/G/c$ FCFS queue in the number of customers in the system. Let N^{RA} be the number of customers served in a busy period of the coupled RA model, then

$$\mathbb{E}(N) \leq \mathbb{E}(N^{RA}).$$

The $M/G/c$ RA model is a combination of c independent $M/G/1$ FCFS queues, any of which is empty a fraction $1-\rho$ of the time. So the fraction of time when all c servers of the RA model are simultaneously idle is $(1-\rho)^c$.

Due to the Poisson Arrivals See Time Averages (PASTA) property, this means that one expects the $M/G/c$ RA model to be found empty by $(1-\rho)^c$ of the arrivals. It means

$$\lim_{m\to\infty} \frac{m}{\sum_{i=1}^{m} N_i^{RA}} = (1-\rho)^c \Rightarrow \lim_{m\to\infty} \frac{\sum_{i=1}^{m} N_i^{RA}}{m} = (1-\rho)^{-c},$$

so by ergodicity of the stable $M/G/c$ FCFS queue, so long as $\rho < 1$, $\mathbb{E}(N^{RA}) = (1-\rho)^{-c}$ and the result holds true. □

3.5.3 Upper bound of carryover probability

As defined before, the number of busy cycles in block $\mathbb{C}_{-j},\ j = 1, 2, \ldots$, is specified by m_{-j}. Denote the maximum service duration in the i^{th} busy cycle in this block as $B^*(i),\ i = 1, \ldots, m_{-j}$, so

$$\mathbb{B}^*_{-j} = \max\{B^*(i), i = 1, \ldots, m_{-j}\}.$$

Because the $B^*(i)$'s are identically distributed, and m_{-j} is deterministic, it is easy to verify that

$$\Pr\left[\mathbb{B}^*_{-j} > x\right] \leq m_{-j} \Pr(B^* > x), \tag{3.18}$$

where B^* is the generic one of $B^*(i)$'s.

Thus the upper bound of $\Pr(\mathcal{E}_{-j})$ (as shown in inequality (3.15)) becomes

$$\begin{aligned}
\Pr(\mathcal{E}_{-j}) &= \Pr\left[\Omega_{-(j+1)} > \mathcal{G}_{-j}\right] \\
&< \Pr\left[(c-1)\mathbb{B}^*_{-(j+1)} > \mathcal{G}_{-j}\right], \text{ (see Proposition 3.5.2)} \\
&= \int_0^\infty \Pr\left[(c-1)\mathbb{B}^*_{-(j+1)} > x \,\middle|\, \mathcal{G}_{-j} = x\right] dF_{G_{-j}}(x) \\
&\leq \mathbb{E}\left(m_{-(j+1)} \Pr\left[B^* > \frac{\mathcal{G}_{-j}}{c-1}\right]\right), \text{(by inequality (3.18) and } \mathbb{B}^*_{-(j+1)} \perp \mathcal{G}_{-j}) \\
&\leq m_{-(j+1)}\mathbb{E}(N)\mathbb{E}\left[\overline{G}\left(\frac{\mathcal{G}_{-j}}{c-1}\right)\right], \text{ (see Proposition 3.5.3)} \\
&\leq (1-\rho)^{-c} m_{-(j+1)}\mathbb{E}\left[\overline{G}\left(\frac{\mathcal{G}_{-j}}{c-1}\right)\right], \text{ (see Proposition 3.5.4)}
\end{aligned}$$

where $\perp$ indicates the relationship of being independent.

Therefore the upper bound of the carryover probability can be represented as

$$\Pr(\mathcal{E}) \leq \sum_{j=1}^{\infty} \Pr(\mathcal{E}_{-j}) < (1-\rho)^{-c} \sum_{j=1}^{\infty} m_{-(j+1)}\mathbb{E}\left[\overline{G}\left(\mathcal{G}^*_{-j}\right)\right], \tag{3.19}$$

where we let $\mathcal{G}^*_{-j} = \frac{\mathcal{G}_{-j}}{c-1}$.

We are going to discuss the upper bound shown in inequality (3.19) in separate implementations for the cases of light tail or heavy tail service duration distributions. We define the term "light tail distribution" in the sense of Asmussen and Glynn [6, p. 163], i.e. those which decay at an exponential rate or faster. Heavy tail distributions as those which have superexponential tails.

The case of light tail service duration distributions

It is easy to verify (see equation (3.14)) that

$$\mathcal{G}^*_{-j} \sim \text{Gamma}(m_{-j}, (c-1)\lambda),$$

and its Laplace-Stieltjes Transform (LST)

$$\mathbb{E}\left(e^{-s\mathcal{G}^*_{-j}}\right) = \left[\frac{(c-1)\lambda}{(c-1)\lambda + s}\right]^{m_{-j}}, \; s > 0.$$

Employing ideas shown in the proof of Chernoff's inequality in [25], for some appropriate

$t > 0$, we have

$$\begin{aligned}
\mathbb{E}\left[\overline{G}\left(\mathcal{G}^*_{-j}\right)\right] &= \mathbb{E}\left(\Pr\left[B > \mathcal{G}^*_{-j} \middle| \mathcal{G}^*_{-j}\right]\right) \\
&= \mathbb{E}\left(\Pr\left[e^{tB} > e^{t\mathcal{G}^*_{-j}} \middle| \mathcal{G}^*_{-j}\right]\right) \\
&\leq \mathbb{E}\left(\frac{\mathbb{E}(e^{tB})}{e^{t\mathcal{G}^*_{-j}}}\right) \text{ (due to Markov's inequality)} \\
&= \mathbb{E}\left(e^{tB}\right)\mathbb{E}\left(e^{-t\mathcal{G}^*_{-j}}\right) \\
&= M_B(t)\left[\frac{(c-1)\lambda}{(c-1)\lambda + t}\right]^{m_{-j}},
\end{aligned}$$

where $M_B(t)$ is the moment generating function (m.g.f.) of the generic service duration B.

As a result, inequality (3.19) becomes

$$\Pr(\mathcal{E}) < (1-\rho)^{-c} M_B(t) \sum_{j=1}^{\infty} m_{-(j+1)} \left[\frac{(c-1)\lambda}{(c-1)\lambda + t}\right]^{m_{-j}}. \tag{3.20}$$

To provide the required upper bound ϵ on $\Pr(\mathcal{E})$, the right hand side of (3.20) should be small. Many choices of t and sequences m_{-j} will achieve this; in particular if $m_{-j} = jm_{-1}$ we need only choose $t > 0$ and m_{-1} large enough. For example, for large enough m_{-1} the ratios of two consecutive terms in the sum will be less than $1/2$, and the inequality can be simplified further as

$$\Pr(\mathcal{E}) < 4(1-\rho)^{-c} M_B(t) m_{-1} \left[\frac{(c-1)\lambda}{(c-1)\lambda + t}\right]^{m_{-1}}.$$

With a specified error tolerance ϵ and fixed t, the value of m_{-1} is determined as

$$\begin{aligned}
m_{-1}(t) = \min\Bigg\{ & m : m \in \mathbb{N}, 2\left[\frac{(c-1)\lambda}{(c-1)\lambda + t}\right]^{m} < 1/2, \\
& 4(1-\rho)^{-c} M_B(t) m \left[\frac{(c-1)\lambda}{(c-1)\lambda + t}\right]^{m} < \epsilon \Bigg\}
\end{aligned} \tag{3.21}$$

In practice, we will choose t so that it does not inflate $M_B(t)$ too much and m_{-1} need not be too large. See Section 3.5.5 for examples. Usually, $m_{-1}(t)$ is a convex function in t on its valid interval (allowing the existence of the m.g.f.). So we can find the value of t which minimizes $m_{-1}(t)$.

The case of heavy tail service duration distributions

Since $(\mathcal{G}^*_{-j})^{-1}$ has the inverse Gamma distribution, it follows [34, p. 710] that

$$\mathbb{E}\left[\left(\mathcal{G}^*_{-j}\right)^{-k}\right] = \frac{(c-1)^k\lambda^k}{(m_{-j}-1)(m_{-j}-2)\cdots(m_{-j}-k)},\ k = 1,\ldots,m_{-j}-1.$$

Assume there exist at least two moments of the service duration distribution, $\mathbb{E}(B^n), n \geq 2$. Similar to the calculation above, we have

$$\begin{aligned}
\mathbb{E}\left[\overline{G}\left(\mathcal{G}^*_{-j}\right)\right] &= \mathbb{E}\left(\Pr\left[B > \mathcal{G}^*_{-j}\middle|\mathcal{G}^*_{-j}\right]\right)\\
&= \mathbb{E}\left(\Pr\left[B^n > (\mathcal{G}^*_{-j})^n\middle|\mathcal{G}^*_{-j}\right]\right)\\
&\leq \mathbb{E}\left(\frac{\mathbb{E}(B^n)}{(\mathcal{G}^*_{-j})^n}\right),\ \text{(due to Markov's inequality)}\\
&= \mathbb{E}\left(B^n\right)\mathbb{E}\left((\mathcal{G}^*_{-j})^{-n}\right)\\
&= \frac{(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)}{(m_{-j}-1)(m_{-j}-2)\cdots(m_{-j}-n)},
\end{aligned}$$

and

$$\begin{aligned}
\Pr(\mathcal{E}) &< (1-\rho)^{-c}\sum_{j=1}^{\infty} m_{-(j+1)}\frac{(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)}{(m_{-j}-1)(m_{-j}-2)\cdots(m_{-j}-n)}\\
&= (1-\rho)^{-c}(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)\sum_{j=1}^{\infty}\frac{m_{-(j+1)}}{(m_{-j}-1)(m_{-j}-2)\cdots(m_{-j}-n)}.
\end{aligned}$$

To accelerate the convergence of the series in the RHS of the inequality shown above, we set $m_{-j} = 2^{j-1}m_{-1}$, $j = 1, 2, \ldots$. Therefore

$$\begin{aligned}
\Pr(\mathcal{E}) &< \frac{4m_{-1}(1-\rho)^{-c}(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)}{(m_{-1}-1)(m_{-1}-2)\cdots(m_{-1}-n)}\\
&< 4(1-\rho)^{-c}(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)m_{-1}(m_{-1}-n)^{-n}
\end{aligned}$$

and the calculation of the value of m_{-1} proceeds in two steps. For fixed n set

$$m_{-1}(n) = \min\left\{m : m \in \mathbb{N}, m > n, 4(1-\rho)^{-c}(c-1)^n\lambda^n\mathbb{E}\left(B^n\right)m(m-n)^{-n} < \epsilon\right\}. \tag{3.22}$$

and choose m_{-1} from among these. To minimize $m_{-1}(n)$, we do not always choose larger n. For example, all moments exist for Lognormal and Weibull distributions. But $\mathbb{E}(B^n) \to \infty$ as $n \to \infty$. So we can optimize $m_{-1}(n)$ on a finite discrete set $\{2,\ldots,n^*\}$, as there will eventually be a point n^* beyond which the moments grow very quickly, causing m_{-1} to do likewise.

Remark: Equation (3.22) is also applicable to the light tail case, and may be easier to work with than (3.21) if $M_B(t)$ is difficult to evaluate.

3.5.4 Algorithm description

Our algorithm is similar to that of the Σ^KM/G/c APQ with extra busy cycles to control the probability of carryover.

1. Simulate the M/G$_K$/c RA model backwards. Based on Algorithm 7, class numbers need to be generated and paired with the service durations.

   When running the M/G$_K$/c RA model forward, the class numbers of customers found at time 0, if any, are simulated with probabilities proportional to $\rho_k, k = 1, \ldots, K$. See Proposition 3.3.1 for details. For others, they are proportional to $\lambda_k, k = 1, \ldots, K$. Service durations are simulated according to corresponding class numbers. Let CLASSES be the class numbers paired with the individual service durations in SERVICES (see Algorithm 7).

   After adjusting the service duration orders according to the service initiation instants in the reversed M/G$_K$/c RA model looking forward in time, we reassign arrivals to the service durations. The outputs of the simulation run are $-T$ (the most recent empty time in the past of the coupled M/G$_K$/c RA model), and sequences of INSTANTS, SERVICES and CLASSES, with corresponding entries being paired.

2. Compute m_{-1} according to equations (3.21) or (3.22). Then generate m_{-1} consecutive busy cycles ending at $-T$, recording arrival instant and associated service duration and class number for each customer. The class number is generated with probabilities proportional to $\lambda_k, k = 1, \ldots, K$, then the service duration is simulated accordingly.

   These m_{-1} busy cycles are generated from the coupled RA model. Starting from the empty state at time 0, we simulate the M/G$_K$/c RA model forward for one cycle. Then we adjust the service duration orders as in the previous step to construct the coupled M/G$_K$/c FCFS queue and count the number of busy cycles in it. If the number is less than m_{-1}, simulate additional RA cycles until the summed number of FCFS cycles is at least m_{-1}. These busy cycles are all shifted backwards so that they end at time $-T$. Denote by $-T_1$ the starting time of these extra busy cycles after the shifting, and insert the detailed records (arrival instants, associated service durations and class numbers) to the beginnings of INSTANTS, SERVICES and CLASSES respectively.

3. Generate a pseudo-customer of class $\mathbb{K}$ arriving at time 0. From time $-T_1$, apply the APQ discipline to the outcomes in Step 2 of this algorithm (the augmented INSTANTS, SERVICES

and CLASSES) and restore the $\Sigma^K\mathrm{M}/\mathrm{G}_K/c$ APQ. Output the service initiation instant of the pseudo-customer. With probability no less than $1-\epsilon$ the output will be a draw from the stationary waiting time of class $\mathbb{K}$ in this system.

This step is implemented with small changes to the algorithm of restoring the $\Sigma^K\mathrm{M}/\mathrm{G}/c$ APQ (Section 3.4.3). The differences are follows.

- Because the distributions of service durations are different for various classes of customers, service durations are also grouped into K sequences: $B^{(k)}(j)$, where $k = 1,\ldots,K$, $j = 1,\ldots,N^{(k)}$, and $N^{(k)} \geq 1$, corresponding to the K-class arrival instants $\{t^{(k)}(j)\}$.
- We initialize with $s_0^{(j)} = -T_1$, $j = 1,\ldots,c$, where $-T_1$ is the beginning of the extra m_{-1} busy cycles.
- In the last step of the loop, replace B_i with $B^{(\kappa_i)}\left(n_{i-1}^{(\kappa_i)}\right)$, i.e. for a customer being selected for service with class number κ_i, the paired service duration will be used. Whenever $n_i^{(k)} > N^{(k)}$, another class k customer will be simulated using the class-specific arrival rate and service duration.

Remarks about this algorithm:

(1) If there is no carryover, Proposition 3.5.1 shows that the target queue is in steady state at time 0. The pseudo-customer is allowed due to the PASTA property. See the argument of the remark in Section 3.4.2.

(2) The $-T_1$ mentioned above was generated as the empty time in the past of the $\mathrm{M}/\mathrm{G}_K/c$ RA model. Since it is a upper bound of the coupled (as described in Proposition 3.4.1) $\mathrm{M}/\mathrm{G}_K/c$ FCFS queue, when the FCFS system is empty the RA model need not be so, thus there would be more totally idle periods during $(-T_1, 0)$ than what we need (m_{-1}). However, this will have no effect on the validity of the algorithm.

(3) Since we use the totally idle periods in the successive block to absorb the potential workload excess from the previous one and apply CFTP to conduct the nearly perfect sampling, we would like to call it as CFTP Block Absorption method.

Also note that, if the potential workload excess could be absorbed completely, then the perfect sampling is achieved. In Section 4.3.2 we will see a variant of this method.

3.5.5 Examples

In this section we present some examples to illustrate the nearly perfect sampling of the $\Sigma^K \mathrm{M}/\mathrm{G}_K/c$ APQ under different service duration distribution assumptions. The computation of m_{-1} is the key.

We consider 2-class and 2-server systems, and the common parameters are specified as follows:

$$\lambda_1 = 1.08, \lambda_2 = 0.72, \mu_1 = 1.2, \mu_2 = 0.8, b_1 = 1, b_2 = 0.5 \text{ and } \epsilon = 10^{-10}.$$

The corresponding occupancy is $\rho = 0.9$, and overall average service duration is 1.

- **Gamma distributions**

  Assume the distribution of class k has shape parameter α_k. The rate parameters are determined as

  $$\theta_k = \alpha_k \mu_k, k = 1, 2$$

  so the m.g.f. of the mixed service duration distribution is

  $$M_B(t) = \sum_{k=1}^{2} \frac{\lambda_k}{\lambda} \left(\frac{\theta_k}{\theta_k - t} \right)^{\alpha_k},$$

  where $0 < t < \min\{\theta_1, \theta_2\}$.

  Given t, the required value of m_{-1} can be found with equation (3.21). Then minimizing m_{-1} with regard to t yields the optimal m_{-1}.

  When $\alpha_1 = \alpha_2 = 1$, the service distributions become exponential. We have $m_{-1} = 102$ when $t = 0.7772$.

  If $\alpha_1 = \alpha_2 = 3$, then $m_{-1} = 50$ when $t = 2.1636$.

- **Weibull distributions**

  The c.d.f. of service durations are specified as

  $$G_k(x) = 1 - e^{-(x/\theta_k)^{\alpha_k}}, k = 1, 2.$$

  To match the service rates, we have

  $$\theta_k = \frac{1}{\mu_k \Gamma(1 + 1/\alpha_k)},$$

where $\Gamma(x)$ stands for the standard Gamma function. For light tailed Weibull distributions, we still choose equation (3.22), because it is a general solution regardless of the tail behaviors of service durations, and it avoids the tricky computation of the moment generating function in (3.21). The n^{th} moment of the service distribution is

$$\mathbb{E}(B^n) = \sum_{k=1}^{2} \frac{\lambda_k}{\lambda} \theta_k^n \Gamma(1 + n/\alpha_k).$$

Let $\alpha_1 = \alpha_2 = 2$, which corresponds to a light tail case. We have the optimal $m_{-1} = 46$ when $n = 23$.

If $\alpha_1 = \alpha_2 = 0.5$, then we have a heavy tail case. It follows that $m_{-1} = 1666$ when $n = 19$.

- **Lognormal distributions**

  In this case, $\log(B^{(k)}) \sim \mathcal{N}(\nu_k, \alpha_k^2), k = 1, 2$. To match the service rates, we have

  $$\nu_k = \log\left(\frac{1}{\mu_k}\right) - \frac{\alpha_k^2}{2}.$$

  The n^{th} moment of service distribution becomes

  $$\mathbb{E}(B^n) = \sum_{k=1}^{2} \frac{\lambda_k}{\lambda} e^{n\nu_k + n^2\alpha_k^2/2}.$$

  Let $\alpha_1 = \alpha_2 = 2$, then $m_{-1} = 7.14 \times 10^7$ when $n = 5$.

  If $\alpha_1 = \alpha_2 = 1$, then $m_{-1} = 7564$ when $n = 9$.

  If $\alpha_1 = \alpha_2 = 0.5$, then $m_{-1} = 132$ when $n = 16$.

- **Pareto distributions**

  The service durations have Pareto distributions given as below:

  $$G_k(x) = 1 - \left(\frac{\theta_k}{x + \theta_k}\right)^{\alpha_k}, \ k = 1, 2.$$

  The n^{th} moment of service distribution is

  $$\mathbb{E}(B^n) = \sum_{k=1}^{2} \frac{\lambda_k}{\lambda} \frac{\theta_k^n \Gamma(n+1)\Gamma(\alpha_k - n)}{\Gamma(\alpha_k)}.$$

Distribution	**Abbr.**	α_1	α_2	m_{-1} ($\epsilon = 10^{-5}$)	m_{-1} ($\epsilon = 10^{-10}$)	m_{-1} ($\epsilon = 10^{-15}$)
Exponential	EXP	1	1	69	102	135
Erlang	ERL	3	3	35	50	65
Weibull-light	WBL	2	2	34	46	58
Weibull-heavy	WBH	0.5	0.5	760	1666	2890
Lognormal	LGN	1	1	1381	7564	29906
Pareto	PRT	6	6	1370	24247	431071

Table 3.3: Values of m_{-1} with different service distribution assumptions.

Distribution	$Var(B)$	$\widehat{\mathbb{E}(W_1)}$	$sd(\widehat{\mathbb{E}(W_1)})$	**C.I. of** $\mathbb{E}(W_1)$
Exponential	1.08	3.17	0.11	(2.96, 3.38)
Erlang	0.39	2.23	0.08	(2.08, 2.39)
Weibull-light	0.33	2.07	0.07	(1.93, 2.21)
Weibull-heavy	5.25	9.26	0.34	(8.59, 9.94)
Lognormal	1.83	4.04	0.15	(3.75, 4.33)
Pareto	1.60	3.62	0.14	(3.34, 3.89)

Table 3.4: Estimates and 95% confidence interval of waiting times of class 1 customers with 1,000 samples, where $\epsilon = 10^{-10}$.

It is clear that

$$\theta_k = \frac{\alpha_k - 1}{\mu_k},\ k = 1, 2,$$

and only moments $\mathbb{E}(B^n),\ n = 1, \ldots, \lfloor \alpha_k - 1 \rfloor$, exist.

Let $\alpha_1 = \alpha_2 = 3$, then $m_{-1} = 5.4 \times 10^{13}$ when $n = 2$.

If $\alpha_1 = \alpha_2 = 6$, then $m_{-1} = 24247$ when $n = 5$.

If $\alpha_1 = \alpha_2 = 20$, then $m_{-1} = 244$ when $n = 16$.

Typical results are summarized in Table 3.3. The values of m_{-1} are not very sensitive to ϵ, especially for the light tail distributions. By generating 1,000 samples for each case of service duration distribution, with $\epsilon = 10^{-10}$, the estimates (means and standard deviations) of waiting times of class 1 and 2 are listed in Tables 3.4 and 3.5. Corresponding e.c.d.f.'s are plotted in Figure 3.10.

3.5.6 Algorithmic analysis of the nearly perfect sampling

In this section, we will analyze the distance to the target distribution and expected runtime of the nearly perfect sampling algorithm.

Distribution	$Var(B)$	$\widehat{\mathbb{E}(W_2)}$	$sd(\widehat{\mathbb{E}(W_2)})$	**C.I. of** $\mathbb{E}(W_2)$
Exponential	1.08	5.02	0.18	(4.66, 5.37)
Erlang	0.39	3.66	0.12	(3.41, 3.90)
Weibull-light	0.33	3.34	0.12	(3.11, 3.57)
Weibull-heavy	5.25	15.24	0.59	(14.07, 16.41)
Lognormal	1.83	7.32	0.27	(6.78, 7.86)
Pareto	1.60	5.86	0.22	(5.43, 6.29)

Table 3.5: Estimates and 95% confidence interval of waiting times of class 2 customers with 1,000 samples, where $\epsilon = 10^{-10}$.

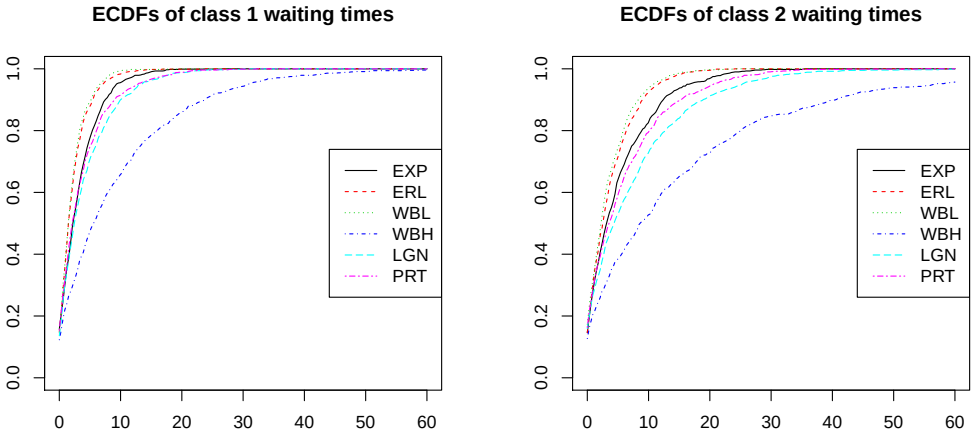

Figure 3.10: The e.c.d.f.'s built of simulations of 1,000 independent draws of waiting times for each class in the APQ using the nearly perfect sampling method, where $\epsilon = 10^{-10}$. The legends of distributions correspond to the abbreviations assigned in Table 3.3.

We will establish that the expected runtime depends upon the occupancy of the queue, the number of servers, and, in the heavy-tailed case, the number of finite moments of the service duration distribution.

Distance to the target distribution

As shown in Section 3.5.3, by simulating extra (m_{-1}) busy cycles of the coupling FCFS queue, we can get steady-state draw of the $\Sigma^K\mathrm{M}/\mathrm{G}_K/c$ APQ with probability more than $1-\epsilon$. When there is carryover of workload excess, the sample at time 0 would underestimate the stationary value, i.e. we are sampling from a statistically smaller distribution with probability less than ϵ.

Proposition 3.5.5 *Let Y be the sample draw of the nearly perfect sampling of the $\Sigma^K M/G_K/c$ APQ in Section 3.5.3, X the steady-state draw of the target system. Then the total variation (v) distance between their distributions is less than ϵ (specified in the algorithm).*

Proof Let C be the event of carryover of workload excess, i.e. given C, $Y \overset{D}{\leq} X$, while in the complement, $Y \overset{D}{=} X$. Let $p = \Pr(C)$. Clearly $p < \epsilon$.

Due to how the algorithm has been defined, $F_Y(\cdot)$ is a mixture of $F_{X|C^c}(\cdot)$ and $F_{Y|C}(\cdot)$:

$$F_Y(x) = (1-p)F_{X|C^c}(x) + pF_{Y|C}(x), \quad x \in E \overset{def}{=} [0,\infty), 0 < p < \epsilon.$$

It is obvious that

$$F_X(x) = (1-p)F_{X|C^c}(x) + pF_{X|C}(x).$$

Therefore according to the definition of total variation (Section 2.1), we obtain

$$\begin{aligned} v &= \frac{1}{2}\int_E |dF_Y(x) - dF_X(x)| = \frac{1}{2}\int_E |pdF_{Y|C}(x) - pdF_{X|C}(x)| \\ &\leq \frac{p}{2}\left(\int_E dF_{Y|C}(x) + \int_E dF_{X|C}(x)\right) = p < \epsilon. \end{aligned}$$

□

Expected runtime

We use the number of customers being generated to quantify the cost in the nearly perfect sampling of the $\mathrm{M}/\mathrm{G}_K/c$ WCQ. By setting the average service duration as 1, it becomes the expected total runtime in the whole system.

There are two procedures that we need to count. The first one is simulating T, the stationary age of the $\mathrm{M}/\mathrm{G}/c$ RA model. In this procedure, there are two steps.

- Simulating the stationary age (ζ) of the M/G/1 FCFS queue in each of these c servers. As mentioned in equation (2.11), $\mathbb{E}(\zeta) = \frac{\mathbb{E}(B^2)}{2\mathbb{E}(B)(1-\rho)^2}$. So the expected runtime counted here has the order of $c(1-\rho)^{-2}$.

- Running c independent M/G/1 FCFS queues until they are idle simultaneously for the first time. As proven in Proposition 3.5.4, $\mathbb{E}(N^{RA}) = (1-\rho)^{-c}$.

The other procedure is generating m_{-1} generic busy cycles of the M/G/c FCFS queue. Based on Proposition 3.5.4, it has upper bound of $m_{-1}(1-\rho)^{-c}$.

The cost of the upper bound in the second procedure is dominant, so we take the expected runtime (upper bound) of this method as

$$m_{-1}(1-\rho)^{-c}. \tag{3.23}$$

In the case of a light tail service duration distribution, based on equation (3.21), it is easy to see that

$$m_{-1} \approx \frac{(c-1)\lambda}{\mu_m}\left[\log\left(\frac{1}{\epsilon}\right) + c\log\left(\frac{1}{1-\rho}\right)\right],$$

where $\mu_m = \min\{\mu_k, k = 1, \ldots, K\}$. Under certain queueing parameter settings, m_{-1} increases linearly with regard to $\log(\frac{1}{\epsilon})$, i.e. $\epsilon \propto e^{-\alpha_0 m_{-1}}$, where α_0 is a constant. It is analogue that the distance limit (ϵ) decays exponentially as m_{-1} increases.

When the service duration distributions are heavy tailed, m_{-1} can be approximated (based on equation (3.22)) as:

$$m_{-1} \approx \min\left\{(c-1)\lambda\left(\frac{1}{\epsilon}\right)^{\frac{1}{n-1}}\left(\frac{1}{1-\rho}\right)^{\frac{c}{n-1}}(\mathbb{E}(B^n))^{\frac{1}{n-1}}, n \geq 2\right\},$$

where $n = 2, 3, \ldots,$ takes appropriate value such that $\mathbb{E}(B^n)$ exists and $\frac{1}{n-1}$ can be fully exerted to depress the value of m_{-1}. As illustrated in the Pareto example in Section 3.5.5, when the service duration distributions have only a finite number n of moments, m_{-1} can be quite large.

If we fix the parameters and only allow ϵ varying, based on the analysis above, the upper bound of the expected runtime has order:

$$\log\left(\frac{1}{\epsilon}\right)(1-\rho)^{-c}, \text{ in the light tail case, and}$$
$$\left(\frac{1}{\epsilon}\right)^{\frac{1}{n-1}}(1-\rho)^{-c}, \text{ in the heavy tail case,}$$

where $n \in \mathbb{N}$ is chosen appropriately such that $\mathbb{E}(B^n)$ is not inflated too much.

Comparison with ordinary simulation

Another way to approach the limiting distribution is performing ordinary simulation: simulating forward from an arbitrarily chosen state, then outputting a draw after some "burn-in" time.

The burn-in time is usually hard to compute. As for the Markov chains mixing exponentially fast, like the GI/G/1 queue with light tail service duration distribution, the relaxation time (γ^{-1}) has been estimated by Asmussen and Glynn [6, p. 101]

$$\gamma = \arg\min_{s>0} \mathbb{E}\left(e^{s(B-A)}\right),$$

where B is the service duration and A inter-arrival time. Therefore the burn-in time is roughly

$$t(\epsilon) \sim \gamma^{-1} \log(1/\epsilon) \text{ as } \epsilon \to 0.$$

Taking the M/M/1 FCFS queue as an example, more precise results were presented by Asmussen [5, p. 107] which controlled the difference of between p.m.f.'s, and by Abate and Whitt [1] which decayed the difference between some of the moments.

However, when the service duration distributions are heavy tailed, these chains could not mix at exponential rates. Abate and Whitt [2] analyzed the transient behavior of the M/G/1 FCFS queue workload process ($W(t)$).

They defined

$$\begin{aligned} m_k(t,x) &= \mathbb{E}\left[W(t)^k | W(0) = x\right], \\ \mathbb{H}_k(t) &= \frac{m_k(t,0)}{m_k(\infty)}, \left(\text{ where } m_k(\infty) = W(\infty)^k\right) \end{aligned}$$

and proved (see Theorem 2 in Abate and Whitt [2]) that $\mathbb{H}_k(\cdot)$ is a proper c.d.f. if the $(k+1)^{st}$ moment of the service distribution exist.

The corollary of Theorem 6 in Abate and Whitt [2] showed

$$\begin{aligned} h_{11} &= \frac{1}{1-\rho}\frac{v_2}{2v_1} \\ h_{12} &= \frac{1}{(1-\rho)^2}\left(\frac{v_3}{3v_1} + v_2\right), \end{aligned}$$

where h_{11} and h_{12} are the first and second moments of distribution $\mathbb{H}_1$, and v_k the k^{th} moment of the stationary unfinished workload.

It is easy to verify that

$$h_{11} = \frac{\mathbb{E}(B^3)}{3(1-\rho)\mathbb{E}(B^2)} + \frac{\rho\mathbb{E}(B^2)}{2(1-\rho)^2\mathbb{E}(B)}.$$

Consider the case of a Pareto service time distribution with shape parameter $\alpha = 3$. Such a distribution would have a mean and a variance, but no third moment. Hence h_{11} does not exist. Therefore, $H_1(t)$ will be a well-defined c.d.f. without a well-defined mean, and the convergence rate of $m_1(t, 0)$ to $m_1(\infty)$ cannot be exponential, and therefore results similar to [1] cannot be achieved.

So if the service duration distributions only have limit moments, $\mathbb{H}_1$ could not be light tailed, thus the mixing rate is no longer exponentially rapid. Furthermore, in the multi-server case, the analysis will become more complicated, and the mixing rate is likely to be slower.

Therefore, the advantage of the nearly perfect sampling algorithm is that it achieves sample distribution having clearly specified distance to the target one with one finite runtime, which can be estimated by equations (3.21), (3.22) and (3.23). Specifically, in the light tail case, the distance to the target distribution is also decayed at exponential rate (see equation (3.23) with light tail settings).

Another merit of the nearly perfect sampling is that it can perform equivalently well or even better than the analytical solutions in practical use. For example, many analytical results are provided in LST forms. To get the final results in the time domain, numerical inversions are needed. As mentioned in Section 2.5.3, the commonly used Gaver-Stehfest method can only achieve at most 5 effective digits under currently common computing resources. But with nearly perfect sampling, we can increase the significant digits by generating more samples while specifying an appropriate total variation bound ϵ.

Chapter 4

Sampling time-varying queues

Most classical queueing models are assumed to be time homogeneous. But in the real world, the distributions of inter-arrival times and service durations are generally time dependent. At service stations such as restaurants, bank counters and telecommunication switches, the arrival rates are higher during rush hours than at slack periods. Another possible illustration of time-varying queues comes from the world of transplant queues, since one source of deceased donor organs is due to traffic accidents leading to death. [35] provides some evidence of seasonality in the donor rate, with a notable peak in the summer when there is more traffic on the roads, whereas the tendency in this regard in [47] is much reduced. Usually, these time-varying patterns are repeated daily, weekly, monthly or yearly, due to cyclic human life styles or seasonal environmental conditions. Therefore queues featuring periodically varying arrival or service rates are both realistic and deserving of effective modeling tools to study them.

In this chapter we present algorithms for perfect and nearly perfect samplings of single-server or multi-server queues with periodic Poisson arrivals. The service durations have periodically time-dependent exponential (e.g. $\mathrm{M}_t/\mathrm{M}_t/1$ and $\mathrm{M}_t/\mathrm{M}_t/c$) or homogeneous general (e.g. $\mathrm{M}_t/\mathrm{G}/1$ and $\mathrm{M}_t/\mathrm{G}/c$) distributions. Assuming the cyclic period has length of 1, we construct discrete dominating processes at instant $n \in \{0, \pm 1, \ldots\}$ by coupling the number of arrivals in the cyclic periods.

With regard to the $\mathrm{M}_t/\mathrm{M}_t/1$ FCFS queue, perfect sampling is obtained with the regenerative method [52]. Once again, since the regenerative method has an infinite expected runtime, a nearly perfect sampling algorithm is proposed by using the Block Absorption method as introduced in the previous chapter. Its distance to the target distribution is less than ϵ in the sense of total variation.

As for the $\mathrm{M}_t/\mathrm{G}/1$ FCFS queue, perfect sampling is achieved with dominated CFTP [29]. Since the unfinished workload is invariant under any work-conserving disciplines in the single-server scenario, the perfect sampling of $\Sigma^K\mathrm{M}_t/\mathrm{G}_K/1$ APQ is readily implemented. As for the

$M_t/G/c$ FCFS and APQ, by using the coupled RA model as the upper bound, their perfect samplings are also available.

In the multi-server case, when the service distributions are different for various classes of customers, i.e. $\Sigma^K M_t/G_K/c$ APQ, we do not achieve nearly perfect sampling, because it is hard to estimate the upper bound of the tail probability of the longest service duration in a busy cycle of the $M_t/G/c$ FCFS queue, where the G in this case represents the aggregate of the various G_k's above, taken as a single class of customers.

Recall that Proposition 3.5.3 claims $\Pr(B^* > x) \leq \mathbb{E}(N)\overline{G}(x)$, where B^* is the the longest service duration, and N the number of customers served in a busy cycle of the $M/G/c$ FCFS queue. The proof relies on the regenerative structure of the homogeneous queue, which does not depend on the time. But in the time-varying system, it is time dependent. Only the time points with integral length intervals, at which the system is empty, constitute a sequence of regeneration points for unfinished workload or waiting times [21]. So the starting points of busy cycles of the $M_t/G/c$ FCFS queue do not constitute a regenerative process. Considering that the service durations in such a busy cycle are still correlated, and that classical renewal theory cannot be applied, it is hard to figure out the upper bound of the tail probability: $\Pr(B^* > x)$.

Before going further into the details, we would like to extend the notations based on the definitions in Section 2.2.2.

- Define
$$\bar{\lambda} = \int_0^1 \lambda(t)dt, \text{ and } \bar{\mu} = \int_0^1 \mu(t)dt.$$
To ensure the stability of the $M_t/M_t/1$ FCFS queue, the occupancy is given by
$$\rho = \frac{\bar{\lambda}}{\bar{\mu}} < 1.$$
As for the $M_t/M_t/c$ FCFS queue ($c \geq 2$), this condition becomes
$$\rho = \frac{\bar{\lambda}}{c\bar{\mu}} < 1.$$
We distinguish the case $c \geq 2$ to underscore that the multi-server queue analysis is usually quite different to that of the single-server case.
As for the $M_t/G/1$ and $M_t/G/c$ FCFS queue, we define:
$$\rho = \frac{\bar{\lambda}}{\mu} \text{ and } \rho = \frac{\bar{\lambda}}{c\mu},$$
where $\mu = 1/\mathbb{E}(B)$. To keep the systems stable, it should be that the occupancy is strictly

less than 1.

- Define

$$F_\lambda(t) = \frac{\int_0^t \lambda(s)ds}{\bar{\lambda}}, \text{ and } F_\mu(t) = \frac{\int_0^t \mu(s)ds}{\bar{\mu}} \tag{4.1}$$

for $t \in (0, 1)$. These functions are strictly increasing on the defined interval, since $\lambda(t)$ and $\mu(t)$ are both positive except at some discrete points (see Section 2.2.2). Therefore their inverse functions do exist, and are denoted by $F_\lambda^{-1}(x)$, and $F_\mu^{-1}(x)$, $x \in (0, 1)$ respectively.

- As for the time-varying queues, let N_k^A be the number of arrivals on interval $(k-1, k], k \in \mathbb{Z}$, so

$$N_k^A \sim \text{Poi}(\bar{\lambda}).$$

The N_k^A's constitute an i.i.d. sequence of r.v.'s.

- In the $M_t/M_t/1$ and $M_t/M_t/c$ FCFS systems, let N_k^D be the number of "potential departures" on interval $(k-1, k], k \in \mathbb{Z}$, assuming the system keeps busy. In the single-server system

$$N_k^D \sim \text{Poi}(\bar{\mu}).$$

In the multi-server case, N_k^D represents the number of potential departures on interval $(k-1, k], k \in \mathbb{Z}$, assuming all c servers are kept continually busy. So

$$N_k^D \sim \text{Poi}(c\bar{\mu}).$$

N_k^D's also constitute an i.i.d. sequence of r.v.'s and they are independent of N_k^A's.

When any server is idle, we ignore the occurrence of the potential departure events in that server.

- In the algorithms that follow we will couple homogeneous queues to the time-varying queues that we are studying. Denote by Q_t^N and Q_t^H the numbers of customers at time t in the time-varying and homogeneous queues respectively. They are defined to be right continuous. At the arrival or departure instants, $Q_t = Q_{t+}$.

4.1 Perfect and nearly perfect sampling of $M_t/M_t/1$ FCFS queue

Since the perfect and nearly perfect sampling of the $M_t/M_t/1$ FCFS queue relies on the backward simulation of the M/M/1 FCFS queue, we start by presenting this algorithm.

4.1.1 Backwards simulating the M/M/1 FCFS queue with a specified number of busy cycles

Assume the arrival and service rates of the M/M/1 FCFS queue are constants λ_0 and μ_0 respectively. Since it is time reversible, as mentioned in Section 2.3.4, conceptually this algorithm is quite straightforward. Multiple busy cycles are simulated for the use by the CFTP Block Absorption method.

In the forward simulation procedure: let E indicate the event of arrival ($E = 1$) or potential departure ($E = -1$) and Q the number of customers in the system. For the record of (t, E, Q), t is the event instant, E the event type and Q the number of customers just before this event. The unused potential departure events are identified by $Q = 0$ and $E = -1$, as these would correspond to the potential departures from an idle queue.

Just before a potential departure event ($E = -1$) occurs, if $Q = 1$, then this departure instant is the end of a busy cycle when simulating forward. To achieve m idle periods after being reversed, these events should be counted for m times after the queue becomes idle for the first time.

When reversing the event instants generated in the forward simulation, usually we only treat the (forward) departures as arrivals and (forward) arrivals as departures. But for the coupling of time-varying queue, since we need the potential departure events, the unused ones will be kept, and they are still treated as unused potential departure events after the reversing.

Algorithm 9 below presents the corresponding pseudocode, where $-T$ is the initial instant of m consecutive busy cycles before time 0 of the backwards simulated M/M/1 FCFS queue, Instants and Events are the event information. Instants stands for the vector of event instants and Events the event types: 1 for arrival and -1 potential departure.

4.1.2 Perfect sampling of $M_t/M_t/1$ FCFS queue with $\inf \mu(t) > \sup \lambda(s)$

Let $\mu_l = \inf \mu(t)$ and $\lambda_u = \sup \lambda(t)$. Assume

$$\mu_l > \lambda_u.$$

Algorithm 9 M/M/1 FCFS queue backward simulation with m busy cycles

Initialize vectors INSTANTS, EVENTS and QLENGTHS to empty.
$t \leftarrow 0$ # event instant, initialized as zero
Simulate $Q \sim \text{Geom}(1 - \rho_0)$, where $\rho_0 = \lambda_0/\mu_0$
while $Q > 0$ **do**
  Simulate $X \sim \text{Exp}(\lambda_0 + \mu_0)$
  $t \leftarrow t + X$
  Simulate E from $\{1, -1\}$ with $\Pr(E = 1) = \lambda_0/(\lambda_0 + \mu_0)$
  Append t to INSTANTS, E to EVENTS and Q to QLENGTHS
  $Q \leftarrow (Q + E)^+$
end while
$M \leftarrow 0$ # counter of the busy cycles
while $M < m$ **do**
  Simulate $X \sim \text{Exp}(\lambda_0 + \mu_0)$
  $t \leftarrow t + X$
  Simulate E from $\{1, -1\}$ with $\Pr(E = 1) = \lambda_0/(\lambda_0 + \mu_0)$
  Append t to INSTANTS, E to EVENTS and Q to QLENGTHS
  if $Q = 1$ and $E = -1$ **then**
    $M \leftarrow M + 1$
    $T \leftarrow t$
  end if
  $Q \leftarrow (Q + E)^+$
end while
Bind INSTANTS,EVENTS and QLENGTHS
Change the signs of EVENTS except those having QLENGTHS $= 0$ and EVENTS $= -1$
Change the signs of INSTANTS and reverse the orders of INSTANTS and EVENTS
return $-T$, INSTANTS and EVENTS

A stable M/M/1 FCFS queue can be generated with arrival and service rates of μ_l and λ_u respectively, since $\mu_l > \lambda_u$. Based on its homogeneous arrival and potential departure events, the time-varying inputs of the coupled $M_t/M_t/1$ queue are simulated as follows:

- The time-varying arrival events are filtered from the homogeneous arrivals by using the thinning method (Section 2.5.2), because we have used too high an arrival rate.

- Time-varying potential departure events are reproduced based on the homogeneous ones by inserting the time-varying potential departure events, because we have used too low a departure rate. The extra time-varying process, which has rate of $\mu(t) - \mu_l$, is generated with the inter-event time method (Section 2.5.2). Therefore the whole potential departure process is a superposition of the homogeneous and the time-varying ones, due to the aggregation property of independent Poisson processes.

Under the coupling scheme described above, it is easy to see that the coupled M/M/1 FCFS queue dominates the time-varying one in Q_t (the number of customers in the system), because the arrivals in the time-varying queue are a subset and the potential departures a superset of the M/M/1 FCFS queue.

Conceptually, we start the homogeneous queue (dominator) and the coupled time-varying queue infinitely long ago. At time 0, both of them are in steady state. In a past time $-T \in \mathbb{R}$, if $Q^H_{-T} = 0$, it means the coupled time-varying queue must be empty at this time and coalescence is achieved. By running it forward with the time-varying events generated as above, we get a steady-state draw of the time-varying queue at time 0. Actually, the argument of Proposition 2.3.1 can be applied here by allowing the target system ($\{X_t\}_{t\in\mathbb{R}}$) to be a time-varying queue.

This algorithm is described as follows:

1. We simulate backwards the M/M/1 FCFS queue with parameters of λ_u and μ_l until it becomes idle at time $\tau \in \mathbb{R}$, where $\tau = \sup\{t : t \in \mathbb{R}, t \le 0, Q^H_t = 0\}$. See Algorithm 9 for reference, with the adjustment that we can stop when the system becomes idle for the first time. We have the homogeneous events recorded in INSTANTS and EVENTS on interval $[\tau, 0)$.

   If $\tau = 0$, return $Q^N_0 = 0$. Otherwise, continue.

2. Select from INSTANTS where EVENTS $= 1$ as ARRIVALS. Filter the arrival instants with the thinning method and get the time-varying arrival instants as ARRIVALS^N.

3. Select from INSTANTS where EVENTS $= -1$ as FULLDEPARTURES. Then add potential extra departures according to the specified $\mu(t)$ as follows.

Let $t_0 = \tau$, and repeat the two sub-steps below for $n = 0, 1, \ldots, N$, where N satisfies $t_N = \max\{t_k : t_k < 0, k \in \mathbb{Z}\}$.

(1) Simulate ξ_{t_n} from the distribution with c.d.f.

$$F_{t_n}(x) = 1 - e^{-\int_0^x [\mu(t_n+s)-\mu_l]ds},$$

where the subscript of F indicates that it depends on t_n.

(2) Assign $t_{n+1} = t_n + \xi_{t_n}$.

If $N > 0$, then append $\{t_1, \ldots, t_N\}$ to FULLDEPARTURES′; otherwise, nothing would be added. Sort all the elements in FULLDEPARTURES′ in ascending order as the time-varying events FULLDEPARTURES^N.

4. Starting from empty state at time τ with inputs ARRIVALS^N and FULLDEPARTURES^N, we run the time-varying system forward until time 0 and output the state Q_0^N as a steady-state draw at the integral times for the $M_t/M_t/1$ FCFS queue.

4.1.3 Perfect sampling of the $M_t/M_t/1$ FCFS queue

In this section, we use the regenerative method to perform perfect sampling of the $M_t/M_t/1$ FCFS queue with general stationary condition, i.e. $\bar{\lambda} < \bar{\mu}$.

The key is to construct a dominating process. Based on the time-varying system, if we concentrate all arrivals to the end of the interval $(k-1, k], k \in \mathbb{N}$, and all potential departures to the beginning of it, the modified process would dominate the original one. Intuitively, since more potential departure events might be "wasted" due to the postponing of arrival events, so there would be more customers remaining in the system. This idea is explicitly stated by the following proposition.

Proposition 4.1.1 *Construct a process by modifying a stable $M_t/M_t/1$ FCFS queue as follows. On each interval of $(k-1, k], k \in \mathbb{Z}$, let the number of arrivals in the $M_t/M_t/1$ queue be N_k^A, and the number of potential departures be N_k^D. In the modified queue, let N_k^A customers arrive as a batch just before time k, and let N_{k+1}^D potential departures occur just after time k. Denote by L_k the number of customers counted at time k of the modified process, and by Q_k^N that in the corresponding $M_t/M_t/1$ FCFS queue. If $L_{k_0} = Q_{k_0}^N = 0$ for some $k_0 \in \mathbb{Z}$, then the modified system dominates the original one in the number of customers at all integer times after k_0:*

$$L_k \geq Q_k^N, \ \forall k \geq k_0.$$

Proof At the non-integer points, we define

$$L_t = \left(L_{k-1} - N_k^D\right)^+, \forall t \in (k-1, k). \tag{4.2}$$

It is obvious that

$$L_k \geq N_k^A, \forall k \in \mathbb{Z},$$

since no matter what the system's state is, the N_k^A arrivals are guaranteed.

It is clear that when $k = k_0$ the inequality is true. Assume when $k = m, m \geq k_0, m \in \mathbb{Z}$, the inequality holds, then when $k = m + 1$ we have:

1. If $\exists t \in (m, m+1), \ni Q_t^N = 0$, then $Q_{m+1}^N \leq N_{m+1}^A \leq L_{m+1}$.

2. Otherwise, $Q_t^N > 0, \forall t \in (m, m+1)$, i.e. the time-varying queue keeps busy on this interval, then

$$Q_{m+1}^N = Q_m^N - N_{m+1}^D + N_{m+1}^A.$$

(1) If $L_t > 0, \forall t \in (m, m+1)$, then

$$L_{m+1} = L_m - N_{m+1}^D + N_{m+1}^A,$$

and it follows that

$$L_{m+1} - Q_{m+1}^N = L_m - Q_m^N \geq 0.$$

(2) Otherwise $\exists t \in (m, m+1), \ni L_t = 0$, then it must be that

$$L_m \leq N_{m+1}^D \Rightarrow Q_m^N \leq N_{m+1}^D.$$

So

$$Q_{m+1}^N = Q_m^N - N_{m+1}^D + N_{m+1}^A \leq N_{m+1}^A \leq L_{m+1}.$$

The mathematical induction principle establishes the result. □

It is clear that $\{L_k\}_{k\geq 0}$ is a non-delayed regenerative process with $L_k = 0$ as the regenerative setting. So its cycle length can be defined as

$$T = \min\{k : k \geq 1, L_k = 0\}, \tag{4.3}$$

where the initial state $L_0 = 0$. The associated cycle of queue lengths during the busy period is given by

$$C = \{L_k : k = 0, \ldots, T-1\}. \tag{4.4}$$

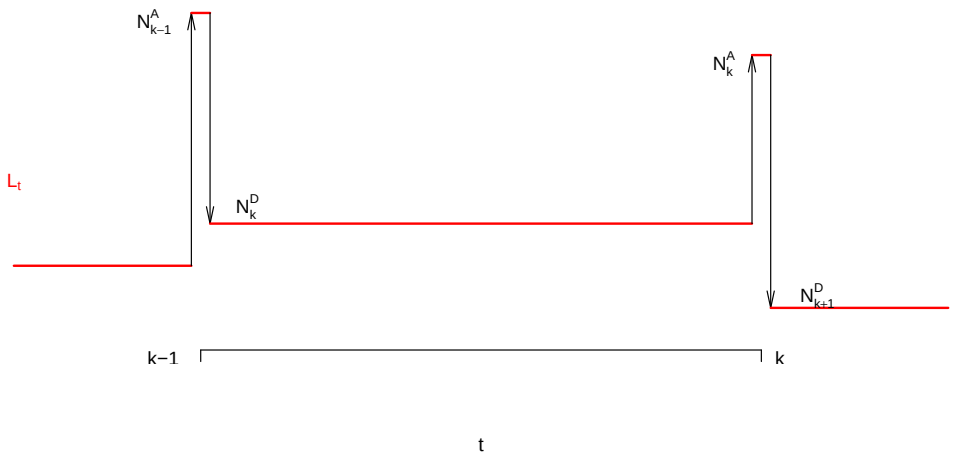

Figure 4.1: Construction of the dominating process of the $M_t/M_t/1$ queue.

According to the definition of L_t (see equation (4.2)), let

$$W_k = L_{k-0.5}, \tag{4.5}$$

we have

$$\begin{cases} W_k = \left(W_{k-1} + N^A_{k-1} - N^D_k\right)^+ \\ L_k = W_k + N^A_k \end{cases}, \tag{4.6}$$

where x^+ means non-negative truncation of x. It is clear that N^A_k is independent of W_k, since W_k is determined by $N^A_i(i < k)$ and $N^D_i(i \le k)$, and N^A_k is independent of these r.v.'s as defined at the beginning of this chapter. A segment path of the dominating process is shown in Figure 4.1. It has some similarity to the discrete queue of LAS (Late Arrival System) [12], but the differences preclude us from using the LAS model directly.

Sampling from the steady-state of the dominating process

The upper formula in Equation (4.6) has the exact form of Lindley's equation of waiting time in a GI/G/1 queue, and it leads to a special perfect sampling algorithm as shown by Asmussen and Glynn [6, p. 437] and Ensor and Glynn [14]; we reproduce the algorithm below.

Let $X_k = N^A_{k-1} - N^D_k$, $k \in \mathbb{N}$. These differences X_k constitute an i.i.d. sequence, which we

generically denote by $X = N^A - N^D$. In light of Equation (4.6) we find

$$W_k = (W_{k-1} + X_k)^+ .$$

Starting from $W_0 = 0$, $S_0 = 0$ and defining $S_k = \sum_{i=1}^{k} X_i, k \in \mathbb{N}$. Then $\{S_k\}_{k\geq 0}$ is a random walk with negative drift (as $\mathbb{E}(X) < 0$), since $\mathbb{E}(N^A_{k-1}) < \mathbb{E}(N^D_k)$. It is shown in [6, p. 3] that

$$W_k \stackrel{D}{=} \max_{i=0,1,\ldots,k} S_i.$$

So the limiting r.v. W_∞, defined by $\lim_{k\to\infty} W_k$, satisfies

$$W_\infty \stackrel{D}{=} \max_{k\geq 0} S_k.$$

To perform Exponential Change of Measure (ECM) (see Section 2.4.2), solve

$$M_X(\gamma) = 1 \tag{4.7}$$

for $\gamma > 0$, where

$$M_X(t) = \mathbb{E}(e^{tX}) = e^{\bar{\lambda}e^t + \bar{\mu}e^{-t} - \bar{\lambda} - \bar{\mu}},$$

so equation (4.7) becomes

$$\bar{\lambda}e^\gamma + \bar{\mu}e^{-\gamma} - \bar{\lambda} - \bar{\mu} = 0.$$

Denote $g(\theta) = \bar{\lambda}e^\theta + \bar{\mu}e^{-\theta} - \bar{\lambda} - \bar{\mu}$. Since $g(0) = 0$, $g'(0) = \bar{\lambda} - \bar{\mu} < 0$, so $\exists\theta^* > 0, \ni\ g(\theta^*) < 0$. Notice that $\lim_{\theta\to\infty} g(\theta) = \infty$, it follows that $g(\theta) = 0$ has positive root on interval (θ^*, ∞). Furthermore, $g''(\theta) = \bar{\lambda}e^\theta + \bar{\mu}e^{-\theta}$, and $g''(\theta) > 0, \forall\theta \in \mathbb{R}$, so it is convex. Therefore the root (γ) of $g(\theta) = 0$ is unique on (θ^*, ∞).

Assume a and d are the observations of N^A and N^D respectively, which are non-negative integers. Let $n = a - d$, so it is an observation of X. Since N^A and N^D are independent, for $n \in \mathbb{Z}$, we have

$$\begin{aligned}
\mathbb{P}_\gamma(X = n) &= \frac{e^{\gamma n}\mathbb{P}(X = n)}{M_X(\gamma)} \\
&= \frac{1}{M_X(\gamma)} \sum_{a,d:a-d=n} e^{\gamma(a-d)}\mathbb{P}(N^A = a)\mathbb{P}(N^D = d) \\
&= \sum_{a,d:a-d=n} \frac{e^{\gamma(a-d)}}{e^{\bar{\lambda}e^\gamma + \bar{\mu}e^{-\gamma} - \bar{\lambda} - \bar{\mu}}} \frac{\bar{\lambda}^a e^{-\bar{\lambda}}}{a!} \frac{\bar{\mu}^d e^{-\bar{\mu}}}{d!} \\
&= \sum_{a,d:a-d=n} \frac{(\bar{\lambda}e^\gamma)^a e^{-\bar{\lambda}e^\gamma}}{a!} \frac{(\bar{\mu}e^{-\gamma})^d e^{-\bar{\mu}e^{-\gamma}}}{d!},
\end{aligned}$$

which means under the measure $\mathbb{P}_\gamma$, X can be treated as the difference of two Poisson r.v.'s: N^{A*} and N^{D*}, which satisfy

$$N^{A*} \sim \text{Poi}(\bar{\lambda}e^{\gamma}) \text{ and } N^{D*} \sim \text{Poi}(\bar{\mu}e^{-\gamma}).$$

So under the measure $\mathbb{P}_\gamma$,

$$\begin{aligned} \mathbb{E}_\gamma(X) &= \mathbb{E}(N^{A*}) - \mathbb{E}(N^{D*}) \\ &= \bar{\lambda}e^{\gamma} - \bar{\mu}e^{-\gamma} \\ &= g'(\gamma). \end{aligned}$$

It is clear that $g'(\gamma) > 0$, due to the convexity of $g(\theta)$ as shown above.

So X (under the measure $\mathbb{P}_\gamma$) can be simulated as below:

- Simulate $N^{A*} \sim \text{Poi}(\bar{\lambda}e^{\gamma})$ and $N^{D*} \sim \text{Poi}(\bar{\mu}e^{-\gamma})$;
- Output $N^{A*} - N^{D*}$.

Under $\mathbb{P}_\gamma$, define a strictly increasing process with ladder heights $S_{\tau(n)}, n = 0, 1, \ldots$, where

$$\tau(0) = 0,\ \tau(n+1) = \inf\{k > \tau(n) : S_k > S_{\tau(n)}\},$$

with $S_0 = 0$.

Let $W = \sup\{S_{\tau(n)} : S_{\tau(n)} \leq V\}$, where $V \sim \text{Exp}(\gamma)$. Then W is a stationary draw of W_∞. Thus the stationary draw of L_∞ is

$$L_\infty = W + N^A,$$

where $N^A \sim \text{Poi}(\bar{\lambda})$, and N^A is independent of W.

Algorithm for perfect sampling of the $\mathbf{M}_t/\mathbf{M}_t/1$ FCFS queue

Based on the constructed dominating process, whose stationary state can be simulated, the perfect sampling of the $\text{M}_t/\text{M}_t/1$ queue is available by using the regenerative method, which can be found in [52], [6, p. 420] and [9].

1. Simulate a random variable (denoted as T^e) from the equilibrium distribution of the cycle length (equation 4.3) of the regenerative process of $\{L_k\}_{k\geq 0}$, which dominates the time-varying queue in queue length at the integer time points.

   As shown in the previous subsection, at time 0, sample a stationary draw of the dominating process, denoted as L_0. Continue simulating it forward until it becomes 0. According

to equation (4.6) we have $L_k = (L_{k-1} - N_k^D)^+ + N_k^A, k \in \mathbb{N}$. So

$$T^e = \min\{k \geq 1, L_k = 0\}.$$

2. Sequentially simulate generic cycles $C^{(j)} = \{L_k^{(j)} : 0 \leq k < T^{(j)}\}, j = 1, 2, \ldots$, of the dominating process, where $T^{(j)}$ is the length of the j^{th} cycle. Record N_k^A and N_k^D $(1 \leq k \leq T^{(j)})$. Stop it when $T^{(J)} \geq T^e$, where

$$J = \min\{j : T^{(j)} \geq T^e\}.$$

3. Using the Order Statistics Method of simulating the time-varying Poisson process (Section 2.5.2) construct time-varying events (arrival and potential departure instants) according to N_k^A and N_k^D $(1 \leq k \leq T_J)$ generated in cycle C_J. Since the cycle length is 1, the corresponding time-varying instant can be computed as (see equation 2.17)

$$t^N = \lfloor t^H \rfloor + F^{-1}(t^H - \lfloor t^H \rfloor),$$

where t^N and t^H stand for the instants in the time-varying and homogeneous systems respectively, and F^{-1} corresponds to the inverse of functions $F_\lambda(t)$ or $F_\mu(t)$ defined in equation (4.1).

From time 0, where the system is empty, simulate forward with these inputs to restore the time-varying queue. Output $Q_{T^e}^N$ as the stationary draw of the number of customers in the $M_t/M_t/1$ queue at time $n \in \mathbb{Z}$.

Remark (1) Proposition 2.4.1 supports that the output $Q_{T^e}^N$ is a steady-state draw.

(2) At time 0, if the stationary draw of L_0 equals zero, we still continue simulating forward.

(3) Since $\{L_k\}_{k\geq 0}$ is the dominating process, we can also directly output $Q_0^N = 0$ if $L_0 = 0$. But in this case, the condition of stopping the generic cycle simulation becomes:

$$J = \min\{j : T^{(j)} > T^e\}.$$

(4) As shown by Proposition 2.4.2, the expected run time of this algorithm is infinite. In the following section, we present a "nearly perfect" sampling algorithm of the $M_t/M_t/1$ queue, which has finite expected runtime.

An example

Let

$$\lambda(t) = 1 + \sin(2\pi t), \quad \mu(t) = 4 + 2\cos(2\pi t).$$

These parameters are the same as those used by Margolius [41], and have $\bar{\lambda} = 1$ and $\bar{\mu} = 4$.

The regenerative method described in the previous subsection is applied for the simulation. On a unit cycle, we choose 100 points at equal spacing from 0 to 1 and generate 10,000 samples for each point. Since only Q_0^N is generated in each trial, to get the samples at different points, we only need to change the phases of the sinusoid functions in each run. So for every point we repeat the algorithm 10,000 times. Although it is time consuming, it shows that our method works quite well.

For a more efficient simulation, we could repeat the algorithm 10,000 times for some fixed phase which is designated as the origin of the clock. Then we could continue simulating the process through a whole cycle. In this case, samples in the simulation would be correlated.

Let us now consider the case where one is interested in the stationary distribution at several time points within one periodic cycle. One alternative, which we have employed here, is that one can repeatedly solve the system of equations for each of the time points of interest corresponding to the reference time 0. An alternative to this approach would be to obtain a stationary draw for the earliest time point of interest, and then continue the simulation forward to the other time point(s) of interest. Thus, a stationary draw at each time point would be obtained, as the queue had already reached stationarity in the first place. Of course, the set of stationary draws thus obtained at the various time points within the cycle would be correlated. For instance, if the initial draw were from a heavily congested queue, then it is likely the later ones would be as well.

The idle probability and expected number of customers at some time $t \in (0, 1)$ of the time-varying queue are illustrated in Figure 4.2. The grey areas indicate the 95% confidence intervals. The time average of Q_t^N is around 0.38288. They match pretty well with the analytical results derived by [41, Section 3], which involve solving a Volterra equation of the second kind numerically.

4.1.4 Nearly perfect sampling of $\mathbf{M}_t/\mathbf{M}_t/1$ FCFS queue

This algorithm is a variant of CFTP. The difficulty of backwards simulation of the dominating process (denoted by $\{L_k\}_{k\in\mathbb{Z}}$, which is the number of customers in it) leads us to resort to a "nearly perfect sampling". A homogeneous queue (M/M/1 FCFS, with $\{Q_k^H\}_{k\in\mathbb{Z}}$ denoting the

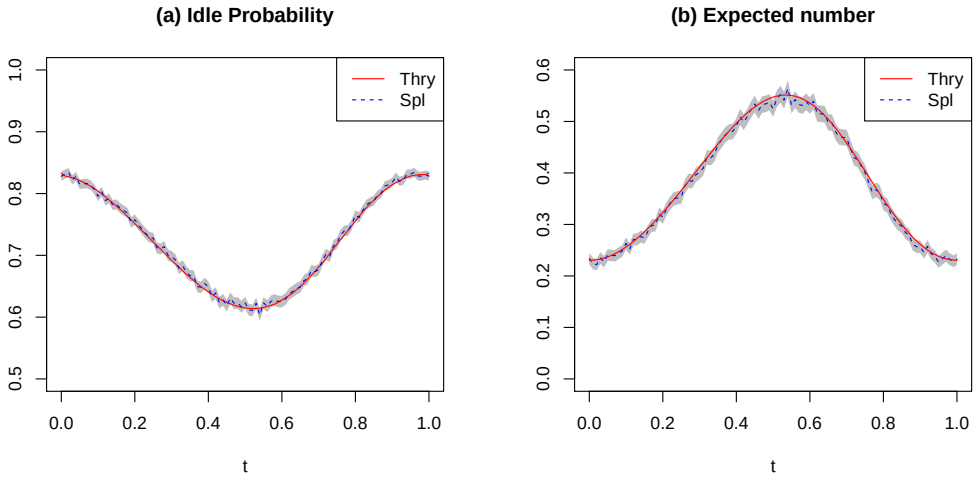

Figure 4.2: Idle probabilities and expected numbers in the $M_t/M_t/1$ queue for one period. 100 points were chosen on it with equal intervals. 10,000 samples were drawn for each point.

number of customers in it) is coupled with the same numbers of arrivals and potential departures on each interval $(k-1,k]$, $k \in \mathbb{Z}$, as that which the time-varying queue has. If $\{Q_k^H\}_{k\in\mathbb{Z}}$ and $\{L_k\}_{k\in\mathbb{Z}}$ are initially empty, Q_k^H and L_k would be close for all $k \in \mathbb{Z}$, because when both systems keep busy on $(k-1,k]$, the updates (i.e. $N_k^A - N_k^D$) of the numbers of customers in both systems are the same. Thus $\{Q_k^H\}_{k\in\mathbb{Z}}$ can be used to approximate the tail of $\{L_k\}_{k\in\mathbb{Z}}$ which was started infinitely long ago.

With regard to consecutive "unit intervals" (which are intervals of $(k-1,k], k \in \mathbb{Z}$) in the homogeneous queue, the coupled time-varying queue might have more customers to be served at the end of these intervals. We call the number of these the "**job excess**". We can compute the upper bound of the job excess when the number of consecutive unit intervals is finite, so it is likely to be consumed by the unused potential departure events in the successive unit intervals. Therefore, if we go backwards sufficiently far and the job excess is consumed completely, then the usual CFTP argument shows that our draw at time 0 is from the limiting distribution. If the job excess goes beyond the number of unused potential departure events, which we refer to as the "**carryover**" of the job excess, the sample would underestimate the steady state. But we can control the probability of carryover to be less than $\epsilon > 0$ by choosing the appropriate number of busy cycles in the homogeneous queue to go backwards.

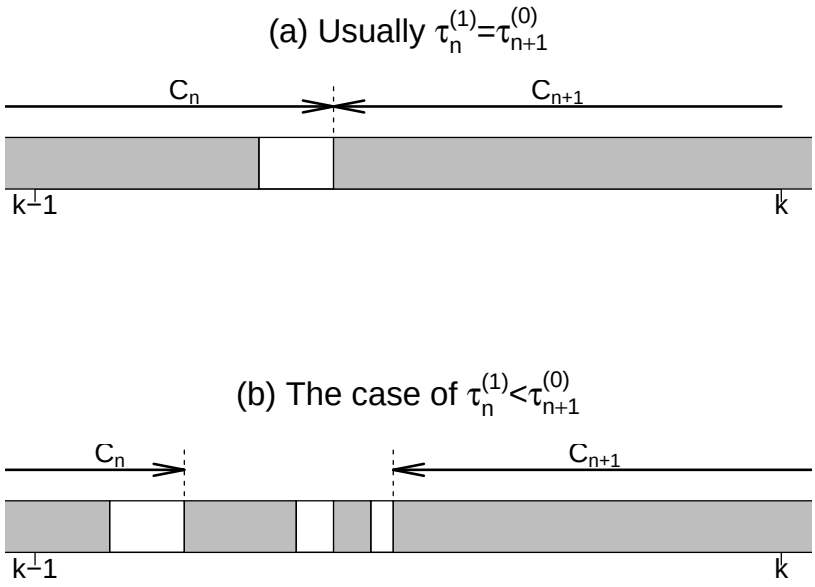

Figure 4.3: Illustrations of the block scheme in the coupled M/M/1 FCFS queue. The gray rectangles are busy periods and blank ones are idle periods. k is an integer. Plot (a) is the usual case. Plot (b) is a possible scenario, where there are some busy cycles between two successive blocks.

Job excess and carryover probability

Denote by $\mathbb{C}_n, n \in \mathbb{Z}$, the block of consecutive m_n busy cycles of the homogeneous queue. It starts at time $\tau_n^{(0)} \in \mathbb{R}$ and ends at $\tau_n^{(1)} \in \mathbb{R}$. We set

$$\begin{cases} \lfloor \tau_n^{(1)} \rfloor = \lfloor \tau_{n+1}^{(0)} \rfloor, \\ \lfloor \tau_n^{(1)} \rfloor \text{ is contained in the last busy cycle of } \mathbb{C}_n. \end{cases} \tag{4.8}$$

The first condition means that the ending of block $\mathbb{C}_n$ and the beginning of $\mathbb{C}_{n+1}$ are located in the same unit interval. The second is set to facilitate the counting of busy cycles in $\mathbb{C}_n$, and it implies that $\lfloor \tau_n^{(0)} \rfloor < \lfloor \tau_n^{(1)} \rfloor$.

It is quite likely that $\tau_n^{(1)} = \tau_{n+1}^{(0)}$. But in an M/M/1 FCFS queue it could happen that one or more busy cycles are contained in a unit interval. When constructing the blocks, it does not matter if we ignore the busy cycles of this type, because they will provide extra unused potential departure events. Please check Figure 4.3 for the illustrations.

Furthermore, let

$$\tau_{-1}^{(1)} = \sup\left\{t : t \le 0, Q_t^H = 0\right\},$$

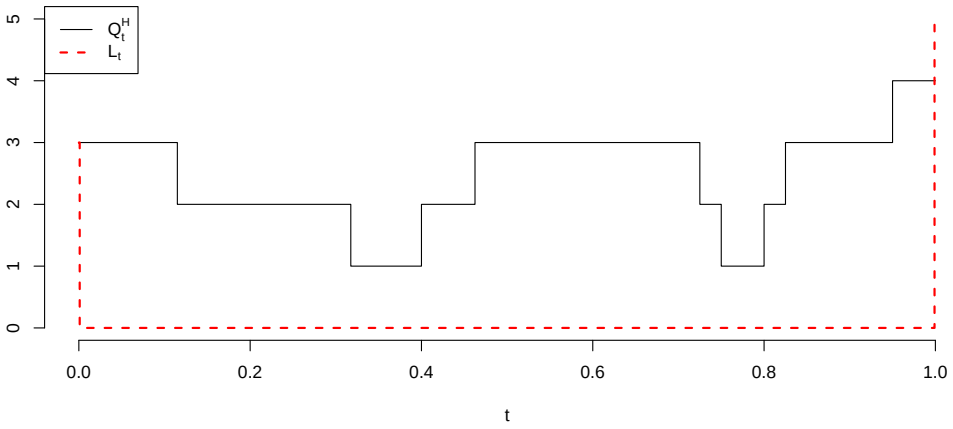

Figure 4.4: The upper bound of individual job excess introduced by constructing the dominating process. Let $\mathbb{C}_1^I = (0, 1]$, $Q_0^H = 3$, $Q_1^H = 4$ and assume there are 5 arrival events (N_1^A) and 4 potential departure events (N_1^D). After rearranging the instants of event occurrence according to the dominating process' construction rule, $L_1 = 5$. So the upper bound of job excess for this interval is $\Omega_1 = L_1 - Q_1^H = 1$.

i.e. if the coupled homogeneous queue is idle at time 0, then block $\mathbb{C}_{-1}$ ends at 0; otherwise, it ends at the most recent idle time before 0.

Define the "involved integral interval" of block $\mathbb{C}_n$ as

$$(\lfloor \tau_n^{(0)} \rfloor, \lfloor \tau_n^{(1)} \rfloor],$$

and denote by $\mathbb{C}_n^I$ this interval. Based on what we specified in equation (4.8), it is clear that the length of $\mathbb{C}_n^I$ takes the natural number $(1, 2, \ldots)$, and $\mathbb{C}_n^I, n \in \mathbb{Z}$, partition the time axis.

The dominating process specified in Proposition 4.1.1 performs as the upper bound of the time-varying queue at integral times. So it is also used to estimate the upper bound of $Q_k^N - Q_k^H, k \in \mathbb{Z}$, which is the job excess at time k.

With $L_{\lfloor \tau_n^{(0)} \rfloor} = Q^H_{\lfloor \tau_n^{(0)} \rfloor}$ and the arrival and potential departure events on interval of $\mathbb{C}_n^I$, we can construct the dominating process as specified in Proposition 4.1.1. The upper bound of the "individual job excess" (which is generated from one block) at the end of $\mathbb{C}_n^I$ is defined as

$$\Omega_n = L_{\lfloor \tau_n^{(1)} \rfloor} - Q^H_{\lfloor \tau_n^{(1)} \rfloor}, \tag{4.9}$$

which is illustrated in Figure 4.4, assuming $n = 1$, $\lfloor \tau_n^{(0)} \rfloor = 0$ and $\lfloor \tau_n^{(1)} \rfloor = 1$.

Since there are m_n idle periods in block $\mathbb{C}_n$, the potential departure events in these periods can be used to absorb the job excess from the previous block. We call the number of these

events the "extra capacity" of $\mathbb{C}_n$, and let $\mathcal{G}_n$ denote it. We have the following proposition about the distribution of the extra capacity of $\mathbb{C}_n$.

Proposition 4.1.2 *The extra capacity in block $\mathbb{C}_n$ has negative binomial distribution with p.m.f as*

$$\Pr(\mathcal{G}_n = k) = \left(\begin{array}{c} k + m_n - 1 \\ k \end{array} \right) p^{m_n}(1-p)^k, k = 0, 1, 2, ...$$

where $p = \frac{\bar{\lambda}}{\bar{\lambda}+\bar{\mu}}$.

Proof Let $\zeta_i, i = 1, 2, ..., m_n$, be the lengths of these idle periods, η_i the numbers of potential departures in the corresponding idle periods in the coupled homogeneous queue. So

$$\mathcal{G}_n = \sum_{i=1}^{m_n} \eta_i.$$

Obviously η_i's are i.i.d.'s and $\zeta_i \sim \text{Exp}(\bar{\lambda})$, $\eta_i|\zeta_i \sim \text{Poi}(\bar{\mu}\zeta_i)$. Therefore we have

$$\begin{aligned} \Pr(\eta_i = k) &= \mathbb{E}(\Pr(\eta_i = k|\zeta_i)) = \mathbb{E}\left(\frac{(\bar{\mu}\zeta_i)^k}{k!}e^{-\bar{\mu}\zeta_i}\right) \\ &= \int_{x=0}^{\infty} \frac{(\bar{\mu}x)^k}{k!}e^{-\bar{\mu}x}\bar{\lambda}e^{-\bar{\lambda}x}dx = \frac{\bar{\lambda}}{\bar{\lambda}+\bar{\mu}}\left(\frac{\bar{\mu}}{\bar{\lambda}+\bar{\mu}}\right)^k, \end{aligned}$$

where $k = 0, 1, 2, \ldots$. It means that

$$\eta_i \sim \text{Geom}(p), \text{ where } p = \frac{\bar{\lambda}}{\bar{\lambda}+\bar{\mu}}.$$

Since $\eta_i, i = 1, 2, ..., m_n$, are independent, their summation has negative binomial distribution, i.e.

$$\mathcal{G}_n \sim \text{NB}(m_n, p).$$

□

Remark In the proof of Proposition 4.1.2, showing η_i has geometric distribution is trivial [18, p. 162, Exercise 8]. We provide this proof to identify the parameter of the geometric distribution clearly.

With Algorithm 9, we can get the arrival and potential departure instants in specified number of busy cycles of the M/M/1 FCFS queue. With these inputs, on each unit interval, $(k-1, k]$, the time-varying inputs can be generated with equation (2.17). We start constructing the time-varying queue with

$$Q^N_{\lfloor \tau^{(0)}_{-1} \rfloor} = Q^H_{\lfloor \tau^{(0)}_{-1} \rfloor},$$

which is the beginning of the involved integral interval $\mathbb{C}^I_{-1}$, run it forward with the time-varying inputs and output Q^N_0 as a steady-state draw of the time-varying queue.

Let $\mathcal{E}$ be the event that $\Omega_{-2} > \mathcal{G}_{-1}$ with $m_{-2} = \infty$, i.e. there might be carryover of the job excess traversing through interval $(\lfloor\tau^{(0)}_{-1}\rfloor, 0)$ after constructing the time-varying queue. When $\mathcal{E}$ occurs the draw at time time 0 would possibly underestimate the stationary value (because the unabsorbed job excess will increase the queue length at time 0), whereas in the complement it definitely matches it.

For finite m_{-j} ($j \in \mathbb{N}$), we let $\mathcal{E}_{-j}$ be the event $\Omega_{-(j+1)} > \mathcal{G}_{-j}$, i.e. the upper bound of individual job excess from block $\mathbb{C}_{-(j+1)}$ goes beyond the extra capacity in $\mathbb{C}_{-j}$. If none of $\mathcal{E}_{-j}$, $j = 1, 2, \ldots$, occurs then there is no carryover. So exactly as what we did in Section 3.5.1, it follows

$$\begin{aligned} \Pr(\mathcal{E}) &\leq \sum_{j=1}^{\infty} \Pr(\mathcal{E}_{-j}) \\ &= \sum_{j=1}^{\infty} \Pr\left[\Omega_{-(j+1)} > \mathcal{G}_{-j}\right]. \end{aligned} \tag{4.10}$$

About the upper bound of individual job excess we have the following proposition.

Proposition 4.1.3 *The upper bound of individual job excess of the involved integral interval $\mathbb{C}^I_n$ after constructing the time-varying queue is less than the maximum number of arrivals in the unit intervals included in $\mathbb{C}^I_n$, i.e.*

$$\Omega_n \leq \mathbb{A}_n,$$

where $\mathbb{A}_n = \max\left\{N^A_k, k = \lfloor\tau^{(0)}_n\rfloor + 1, \ldots, \lfloor\tau^{(1)}_n\rfloor\right\}$.

Proof By observing W_k, which has been defined in equation (4.5), there are two possible cases:

- $W_k > 0, \forall k = \lfloor\tau^{(0)}_n\rfloor + 1, \ldots, \lfloor\tau^{(1)}_n\rfloor$, which means process $\{L_t\}, t \in \mathbb{R}$, does not become idle on $(\lfloor\tau^{(0)}_n\rfloor, \lfloor\tau^{(1)}_n\rfloor]$. So we have

$$L_{\lfloor\tau^{(1)}_n\rfloor} = Q^H_{\lfloor\tau^{(0)}_n\rfloor} + \sum_{k=\lfloor\tau^{(0)}_n\rfloor+1}^{\lfloor\tau^{(1)}_n\rfloor} (N^A_k - N^D_k).$$

  Note that

$$Q^H_{\lfloor\tau^{(1)}_n\rfloor} = Q^H_{\lfloor\tau^{(0)}_n\rfloor} + \sum_{k=\lfloor\tau^{(0)}_n\rfloor+1}^{\lfloor\tau^{(1)}_n\rfloor} (N^A_k - N^{D*}_k),$$

where N_k^{D*} is the number of departures on $(k-1, k]$. It is obvious that

$$N_k^{D*} \leq N_k^D.$$

According to the definition of the upper bound of individual job excess (equation (4.9)), it follows that

$$\Omega_n = L_{\lfloor \tau_n^{(1)} \rfloor} - Q^H_{\lfloor \tau_n^{(1)} \rfloor} = - \sum_{k=\lfloor \tau_n^{(0)} \rfloor + 1}^{\lfloor \tau_n^{(1)} \rfloor} (N_k^D - N_k^{D*}) \leq 0 \leq \mathbb{A}_n.$$

- The complementary situation is that $\exists k \in \{\lfloor \tau_n^{(0)} \rfloor + 1, \ldots, \lfloor \tau_n^{(1)} \rfloor\}, \ni W_k = 0$, i.e. there exists an unit interval where process $\{L_t\}, t \in \mathbb{R}$, becomes idle on $(\lfloor \tau_n^{(0)} \rfloor, \lfloor \tau_n^{(1)} \rfloor]$.

  - If $W_{\lfloor \tau_n^{(1)} \rfloor} = 0$, which means there exists an idle instant in the last unit interval of process $\{L_t\}, t \in \mathbb{R}$, then the number of customers of it at time $\lfloor \tau_n^{(1)} \rfloor$ is exactly the number of arrivals on this interval, i.e. $L_{\lfloor \tau_n^{(1)} \rfloor} = N^A_{\lfloor \tau_n^{(1)} \rfloor}$. So

$$\Omega_n = L_{\lfloor \tau_n^{(1)} \rfloor} - Q^H_{\lfloor \tau_n^{(1)} \rfloor} \leq L_{\lfloor \tau_n^{(1)} \rfloor} = N^A_{\lfloor \tau_n^{(1)} \rfloor} \leq \mathbb{A}_n.$$

  - Otherwise, there exists an idle interval (assuming it is $(k'-1, k']$) of process $\{L_t\}, t \in \mathbb{R}$, on $(\lfloor \tau_n^{(0)} \rfloor, \lfloor \tau_n^{(1)} \rfloor - 1]$, which is the nearest one prior to $\lfloor \tau_n^{(1)} \rfloor - 1$. Explicitly: $\exists k' \in \{\lfloor \tau_n^{(0)} \rfloor + 1, \ldots, \lfloor \tau_n^{(1)} \rfloor - 1\}, \ni W_{k'} = 0$ and $W_k > 0, \forall k = k'+1, \ldots, \lfloor \tau_n^{(1)} \rfloor$. So

$$L_{k'} = N^A_{k'}.$$

    Because process $\{L_t\}, t \in \mathbb{R}$, keeps busy on $(k', \lfloor \tau_n^{(1)} \rfloor]$, it follows

$$L_{\lfloor \tau_n^{(1)} \rfloor} = L_{k'} + \sum_{k=k'+1}^{\lfloor \tau_n^{(1)} \rfloor} \left(N_k^A - N_k^D \right).$$

    As for the homogeneous queue, we have

$$Q^H_{\lfloor \tau_n^{(1)} \rfloor} = Q^H_{k'} + \sum_{k=k'+1}^{\lfloor \tau_n^{(1)} \rfloor} \left(N_k^A - N_k^{D*} \right),$$

    where N_k^{D*} is the actual number of departures on $(k-1, k]$ as specified before.

Therefore we have

$$\Omega_n = L_{\lfloor\tau_n^{(1)}\rfloor} - Q^H_{\lfloor\tau_n^{(1)}\rfloor} = L_{k'} - Q^H_{k'} - \sum_{k=k'+1}^{\lfloor\tau_n^{(1)}\rfloor}(N_k^D - N_k^{D*}) \le L_{k'},$$
$$\Rightarrow \Omega_n \le L_{k'} = N^A_{k'} \le \mathbb{A}_n.$$

Our claim holds true in both cases. It establishes the result. □

Upper bound of the carryover probability

The involved integral interval, $\mathbb{C}_n^I$, contains $\lfloor\tau_n^{(1)}\rfloor - \lfloor\tau_n^{(0)}\rfloor$ unit intervals. It is clear that $N_k^A, k = 1, \ldots, \lfloor\tau_n^{(1)}\rfloor - \lfloor\tau_n^{(0)}\rfloor$, are not independent any more since they come from the m_n busy cycles of the homogeneous queue, and their distributions also depend on $\lfloor\tau_n^{(1)}\rfloor - \lfloor\tau_n^{(0)}\rfloor$, which is a random variable and is not easy to analyze. So the usual way of estimating the tail probability's upper bound, e.g. using the inequality $\Pr(\mathbb{A}_n > x) \le \mathbb{E}(\lfloor\tau_n^{(1)}\rfloor - \lfloor\tau_n^{(0)}\rfloor)\Pr(N^A > x)$, does not hold.

Similar to what we did in Proposition 3.5.3, the upper bound can be figured out with renewal theory.

Proposition 4.1.4 *Let $\mathbb{A}_n$ be the maximum number of arrivals in the involved integral interval $\mathbb{C}_n^I$, as specified in Proposition 4.1.3. Let $\mathcal{N}$ be the number of arrivals in block $\mathbb{C}_n$. Then the upper bound of the tail probability of the distribution of $\mathbb{A}_n$ is*

$$\Pr(\mathbb{A}_n > x) \le \mathbb{E}(\mathcal{N})\frac{\overline{F}_{N^A}(x)}{1 - e^{-\bar{\lambda}}}, \tag{4.11}$$

where $x \ge 0$, m_n is the number of busy cycles in block $\mathbb{C}_n$, N^A is the number of arrivals on the unit interval and $\overline{F}_{N^A}(x) = \Pr(N^A > x)$.

Proof Denote by $t_i, i = 0, 1, \ldots$, the chronological arrival instants of customers. Construct a regenerative process $\{X_i\}_{i\ge0}$ as:

$$X_i = N_k^A, \text{ for } t_i \in (k-1, k], k \in \mathbb{Z}.$$

The embedded renewal process is the sequence of initial instants of every m_n busy cycles of the M/M/1 FCFS queue.

Only when $N_k^A > 0$, does there exist an arrival on the interval of $(k-1, k]$ to record the information of arrivals' number. So it implies that $X_i = N_k^A|(N_k^A > 0)$. For the writing convenience, let N^{A+} denote $N_k^A|(N_k^A > 0), k \in \mathbb{Z}$.

According to Asmussen [5, Corollary 1.4, p. 171], we have

$$\mathbb{E}_e(f(X_i)) = \frac{1}{\mathbb{E}(\mathcal{N})}\mathbb{E}\sum_{j=0}^{\mathcal{N}-1} f(X_j) \Leftrightarrow \mathbb{E}_e(f(N^{A+})) = \frac{1}{\mathbb{E}(\mathcal{N})}\mathbb{E}\sum_{j=0}^{\mathcal{N}-1} f(X_j)$$

for any measurable $f(\cdot)$ where $\mathbb{E}_e$ corresponds to the stationary (or marginal) measure.

Let $f(y) = \mathbb{1}(y > x)$ be the standard indicator function of an event, then it follows that

$$\mathbb{E}_e(\mathbb{1}(N^{A+} > x)) = \frac{1}{\mathbb{E}(\mathcal{N})}\mathbb{E}\sum_{j=0}^{\mathcal{N}-1} \mathbb{1}(X_j > x). \tag{4.12}$$

Denote by $\mathcal{S}$ the integer set of $\left\{\lfloor\tau_n^{(0)}\rfloor + 1, \ldots, \lfloor\tau_n^{(1)}\rfloor\right\}$. It is clear that

$$\begin{aligned}\mathbb{A}_n = \max\{N_k^A : k \in \mathcal{S}\} &= \max\{N_k^A, k \in \mathcal{S} \text{ and } N_k^A > 0\} \\ &= \max\{X_j, j = 0, \ldots, \mathcal{N}-1\}.\end{aligned}$$

So we have

$$\begin{aligned}\mathbb{1}(\mathbb{A}_n > x) &= \mathbb{1}(\max\{X_j, j = 0, \ldots, \mathcal{N}-1\} > x) \\ &\leq \sum_{j=0}^{\mathcal{N}-1} \mathbb{1}(X_j > x)\end{aligned}$$

After taking expected values of both sides of the inequality shown above and combining equation (4.12), we have

$$\begin{aligned}\Pr(\mathbb{A}_n > x) &= \mathbb{E}(\mathbb{1}(\mathbb{A}_n > x)) \\ &\leq \mathbb{E}\sum_{j=0}^{\mathcal{N}-1} \mathbb{1}(X_j > x) = \mathbb{E}(\mathcal{N})\mathbb{E}_e(\mathbb{1}(N^{A+} > x)) \\ &= \mathbb{E}(\mathcal{N})\Pr(N^{A+} > x).\end{aligned}$$

Obviously, $\Pr(N^{A+} = k) = \frac{\Pr(N^A=k)}{1-\Pr(N^A=0)}, k = 1, 2, \ldots$, and $\Pr(N^A = 0) = e^{-\bar{\lambda}}$. So

$$\Pr(N^{A+} > x) = \frac{\overline{F}_{N^A}(x)}{1 - e^{-\bar{\lambda}}}.$$

Therefore

$$\Pr(\mathbb{A}_n > x) \leq \mathbb{E}(\mathcal{N})\frac{\overline{F}_{N^A}(x)}{1 - e^{-\bar{\lambda}}}.$$

□

Based on Propositions of 4.1.2, 4.1.3 and 4.1.4, we continue to analyze the upper bound of carryover probability specified in inequality (4.10). As for $\mathcal{E}_{-j}$, $j = 1, 2, \ldots$, which is event $\Omega_{-(j+1)} > \mathcal{G}_{-j}$, we know that

$$\Omega_{-(j+1)} \leq \mathbb{A}_{-(j+1)} \text{ and } \mathcal{G}_{-j} \sim \text{NB}(m_{-j}, p),$$

where $p = \frac{\bar{\lambda}}{\bar{\lambda}+\bar{\mu}}$. Since $\mathbb{A}_{-(j+1)} \perp \mathcal{G}_{-j}$, we have

$$\begin{aligned}
\Pr(\mathcal{E}_{-j}) &= \Pr\left[\Omega_{-(j+1)} > \mathcal{G}_{-j}\right] \\
&\leq \Pr\left[\mathbb{A}_{-(j+1)} > \mathcal{G}_{-j}\right] \\
&= \mathbb{E}\left(\mathbb{E}(\mathcal{N})\frac{\overline{F}_{N^A}(\mathcal{G}_{-j})}{1-e^{-\bar{\lambda}}}\right) \\
&= \frac{\mathbb{E}(\mathcal{N})}{1-e^{-\bar{\lambda}}}\mathbb{E}\left[\overline{F}_{N^A}(\mathcal{G}_{-j})\right],
\end{aligned}$$

where $\mathcal{N}$ is the number of arrivals in block $\mathbb{C}_{-(j+1)}$. It is well known (e.g. Kleinrock [32, p. 217]) that the expected number of customers served in a busy cycle of the M/G/1 FCFS queue is $1/(1-\rho)$. Since there are $m_{-(j+1)}$ busy cycles in $\mathbb{C}_{-(j+1)}$, it follows that $\mathbb{E}(\mathcal{N}) = \frac{m_{-(j+1)}}{1-\rho}$. So

$$\Pr(\mathcal{E}_{-j}) \leq \frac{m_{-(j+1)}}{(1-e^{-\bar{\lambda}})(1-\rho)}\mathbb{E}\left[\overline{F}_{N^A}(\mathcal{G}_{-j})\right]. \tag{4.13}$$

By employing ideas shown in the proof of Chernoff's inequality (c.f. Hoeffding [25]), for some appropriate $t > 0$, we have

$$\begin{aligned}
\mathbb{E}\left[\overline{F}_{N^A}(\mathcal{G}_{-j})\right] &= \mathbb{E}\left[\Pr(N^A > \mathcal{G}_{-j}|\mathcal{G}_{-j})\right] \\
&= \mathbb{E}\left[\Pr(e^{tN^A} > e^{t\mathcal{G}_{-j}}|\mathcal{G}_{-j})\right] \\
&\leq \mathbb{E}\left[\frac{\mathbb{E}(e^{tN^A})}{e^{t\mathcal{G}_{-j}}}\right] (\text{ due to Markov's inequality}) \\
&= \mathbb{E}(e^{tN^A})\mathbb{E}(e^{-t\mathcal{G}_{-j}}).
\end{aligned}$$

It is easy to see that the m.g.f. of Poisson r.v. N^A is

$$\mathbb{E}(e^{tN^A}) = e^{\bar{\lambda}(e^t-1)}.$$

The LST of negative binomial r.v. $\mathcal{G}_{-j}$ is

$$\mathbb{E}(e^{-t\mathcal{G}_{-j}}) = \left(\frac{p}{1-(1-p)e^{-t}}\right)^{m_{-j}} = \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m_{-j}}.$$

Therefore we have

$$\Pr(\mathcal{E}_{-j}) \leq \frac{m_{-(j+1)}}{(1-e^{-\bar{\lambda}})(1-\rho)} e^{\bar{\lambda}(e^t-1)} \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m_{-j}}. \tag{4.14}$$

So inequality (4.10) becomes

$$\begin{aligned}\Pr(\mathcal{E}) &\leq \sum_{j=1}^{\infty} \Pr(\mathcal{E}_{-j}) \\ &\leq \frac{e^{\bar{\lambda}(e^t-1)}}{(1-e^{-\bar{\lambda}})(1-\rho)} \sum_{j=1}^{\infty} m_{-(j+1)} \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m_{-j}},\end{aligned}$$

where $t > 0$.

Similar to the argument as we used to simplify inequality (3.20), we choose $m_{-j} = jm_{-1}$, and ensure that

$$\left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m_{-1}} < \frac{1}{4}.$$

by adjusting the value of m_{-1}. Then it follows that

$$\Pr(\mathcal{E}) \leq \frac{4e^{\bar{\lambda}(e^t-1)} m_{-1}}{(1-e^{-\bar{\lambda}})(1-\rho)} \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m_{-1}}.$$

With a specified total variation tolerance ϵ and fixed t, the value of m_{-1} is determined as

$$m_{-1}(t) = \min\left\{m : \frac{4e^{\bar{\lambda}(e^t-1)} m}{(1-e^{-\bar{\lambda}})(1-\rho)} \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m} < \epsilon, \left(\frac{\rho}{\rho+1-e^{-t}}\right)^{m} < \frac{1}{4}, m \in \mathbb{N}\right\} \tag{4.15}$$

In practice, we will choose t such that m_{-1} is an acceptable number, i.e. it would not consume an unreasonably long runtime based on the available computing resources.

Algorithm for nearly perfect sampling of the $\mathbf{M}_t/\mathbf{M}_t/1$ FCFS queue

Based on the analysis above, this algorithm can be described as follows.

1. Compute m_{-1} according to equation (4.15).
2. Starting from time 0, simulate backwards an M/M/1 FCFS queue (Algorithm 9) with arrival rate $\bar{\lambda}$ and service rate $\bar{\mu}$ for m_{-1} busy cycles, whose initial instant is $-T$. It is

obvious that $-T = \tau_{-1}^{(0)}$, which is the beginning of block $\mathbb{C}_{-1}$. Record the arrival and potential departure instants in this procedure.

3. Continue simulating backwards the M/M/1 FCFS queue to time $\lfloor -T \rfloor$. Keep recording the arrival and potential departure events and $Q^H_{\lfloor -T \rfloor}$.

4. With equation (2.17), construct the coupled time-varying arrival and potential departure events with the generated homogeneous events in Steps 2 and 3 on each unit interval from $\lfloor -T \rfloor$ to 0.

5. Starting from $\lfloor -T \rfloor$ with $Q^N_{\lfloor -T \rfloor} = Q^H_{\lfloor -T \rfloor}$ and the coupled time-varying events, run the time-varying queue forward and output Q^N_0 as a stationary draw at time 0 of the $M_t/M_t/1$ FCFS queue.

Remark (1) Proposition 3.5.1 supports that Q^N_0 is a steady-state draw at time 0 if there is no carryover.

(2) Since we do not implement the backward simulation of the dominating process, we only use it to estimate the upper bound of the time-varying queue at some time in the past.

An example

The parameters are the same as those used by Margolius [41] and Zeifman et al. [60]:

$$\lambda(t) = 1 + \sin(2\pi t), \quad \mu(t) = 4 + 2\cos(2\pi t). \tag{4.16}$$

So $\bar{\lambda} = 1$ and $\bar{\mu} = 4$. With equation (4.15), specifying $\epsilon = 10^{-10}$ we have $m_{-1} = 23$, when $t = 1.5387$.

Since only Q^N_0 is generated in each trial, to get the samples at different times in a periodic cycle (whose length is 1), we only need to change the phases of the sinusoid functions in each run.

The idle probability and expected number of customers at some time $t \in (0, 1)$ of the $M_t/M_t/1$ queue are illustrated in Figure 4.5. The gray area indicates the 95% confidence intervals of $Q^N_t, t \in (0, 1)$. They match pretty well with the analytical results derived by Margolius [41].

4.2 Nearly perfect sampling of $M_t/M_t/c$ FCFS queue

In the multi-server case, Proposition 4.1.1 can be generalized as follows.

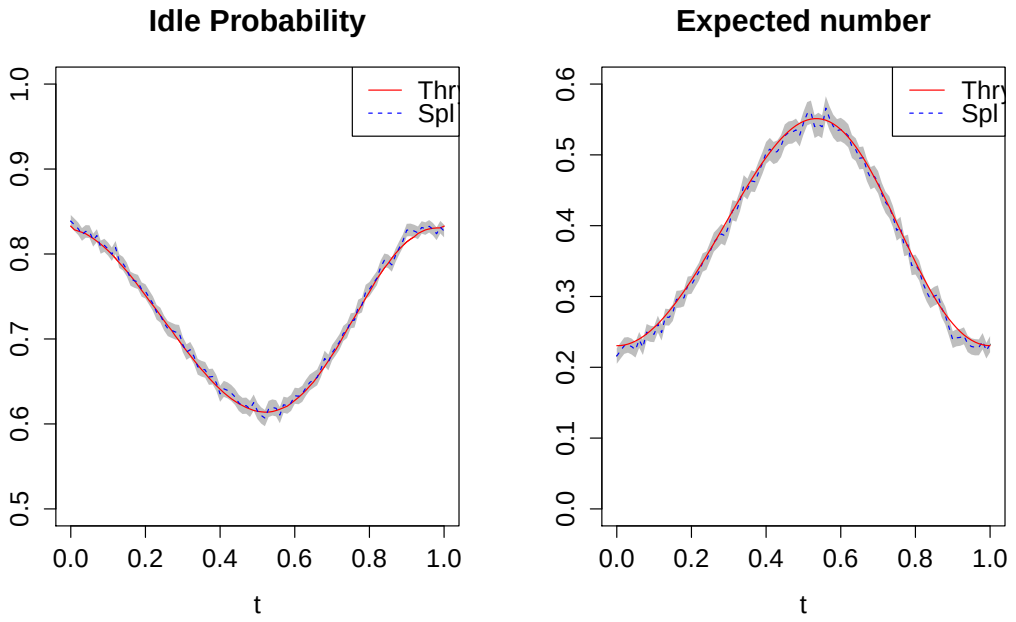

Figure 4.5: Idle probabilities and expected numbers in the $M_t/M_t/1$ queue for one period. 100 points were chosen on it with equal intervals. 10,000 samples were drawn for each point.

Proposition 4.2.1 *Construct $\{L_k\}_{k\in\mathbb{Z}}$ as was done in Proposition 4.1.1. Note that $N_k^D \sim Poi(c\bar{\mu})$ in the multi-server case. If $L_{k_0} = Q_{k_0}^N = 0$ for some $k_0 \in \mathbb{Z}$, then*

$$L_k + c - 1 \geq Q_k^N, \ \forall k \geq k_0.$$

Proof The proof follows exactly the same way as that of Proposition 4.1.1. It is obvious that

$$L_k \geq N_k^A, \forall k \in \mathbb{Z},$$

since no matter what the system's state is, the N_k^A arrivals are guaranteed.

It is clear that when $k = k_0$ the inequality is true. Assume when $k = m, m \geq k_0, m \in \mathbb{Z}$, the inequality holds, then when $k = m + 1$ we have:

1. If $\exists t \in (m, m+1), \ni Q_t^N \leq c - 1$, then $Q_{m+1}^N \leq N_{m+1}^A + c - 1 \leq L_{m+1} + c - 1$.

2. Otherwise, $Q_t^N \geq c - 1, \forall t \in (m, m+1)$, i.e. all servers in the time-varying queue keep busy on this interval, then

$$Q_{m+1}^N = Q_m^N - N_{m+1}^D + N_{m+1}^A.$$

(1) If $L_t > 0, \forall t \in (m, m+1)$, then

$$L_{m+1} = L_m - N^D_{m+1} + N^A_{m+1},$$

and it follows that

$$L_{m+1} - Q^N_{m+1} = L_m - Q^N_m \geq -(c-1) \Rightarrow L_{m+1} + c - 1 \geq Q^N_{m+1}.$$

(2) Otherwise $\exists t \in (m, m+1), \ni L_t = 0$, then it must be that

$$L_m \leq N^D_{m+1} \Rightarrow Q^N_m \leq N^D_{m+1} + c - 1.$$

So

$$Q^N_{m+1} = Q^N_m - N^D_{m+1} + N^A_{m+1} \leq N^A_{m+1} + c - 1 \leq L_{m+1} + c - 1.$$

The mathematical induction principle establishes the result. □

Remark:

(1) The extra number of $c-1$, compared to the upper bound in the single-server system, appearing as part of the dominating process, is caused by the "partly busy" (the number of busy servers is less than c) behaviour of the multi-server system. E.g. there are $c-1$ customers remaining in the system, which have been assigned to some servers. Unfortunately, after the assignment there are no potential departure events in these servers, while there are quite a lot (no less than $c-1$) in the remaining one idle server on interval $(k-1, k]$. So these potential departures are wasted and the upper bound is inflated by $c-1$.

(2) Because the dominating process constructed by Proposition 4.2.1 does not return to zero, we cannot apply the regenerative method to perform the perfect sampling of the $M_t/M_t/c$ FCFS queue. But the CFTP Block Absorption method still works.

4.2.1 Backwards simulating the $M/M/c$ FCFS queue with specified number of busy cycles

Similar to what we did in Section 3.5, we use the $M/M/c$ FCFS queue to couple the time-varying one by ensuring they share the same numbers of arrivals and potential departures on interval $(k-1, k], k \in \mathbb{Z}$.

It is well known that M/M/c FCFS queue is time reversible [49, p. 399]. So this algorithm is similar to what has been done in Algorithm 9. Please check Algorithm 10 for the pseudocode.

Note that in Algorithm 9, we achieved all potential departure events. But the outputted departure events of Algorithm 10 are actual departures. The unused potential departure events will be generated when restoring the $M_t/M_t/c$ FCFS queue through simulating forward.

Assume the arrival and service rates are constants λ_0 and μ_0, which satisfies

$$\rho = \frac{\lambda_0}{c\mu_0} < 1.$$

Denote $p_i = \Pr(Q = i), i = 0, 1, \ldots$, as shown by Kleinrock [32, p. 102]:

$$p_i = \begin{cases} p_0 \frac{(c\rho)^i}{i!}, & i \le c \\ p_0 \frac{\rho^i c^c}{c!}, & i > c, \end{cases} \tag{4.17}$$

where

$$p_0 = \left[\sum_{j=0}^{c-1} \frac{(c\rho)^j}{j!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right]^{-1}.$$

4.2.2 Simulating the potential departure events in each server

By applying Algorithm 10, we get a past time $-T \in \mathbb{R}$, which is the beginning of m busy cycles of the M/M/c FCFS queue such that there are m totally idle periods on interval $(-T, 0)$. Keep going backwards from time $-T$ to simulate the homogeneous queue until time $\lfloor -T \rfloor$. Append the event instants and types to INSTANTS and EVENTS (which are outputted by Algorithm 10) respectively, and record the number of customers at time $\lfloor -T \rfloor$ as $Q^H_{\lfloor -T \rfloor}$. Thus the algorithm of simulating the potential departure events has the inputs as: $Q^H_{\lfloor -T \rfloor}$, INSTANTS (assuming it has n elements) and EVENTS ("1" for arrival and "-1" for departure).

To tell in which server the potential departure events happen, we introduce a vector $\vec{s} = \left(s^{(1)}, \ldots, s^{(c)}\right)$, where $s^{(l)}, l = 1, \ldots, c$, is an indicator variable for the event that the l^{th} server is busy. When any server becomes idle or changes from idle to busy, we record the corresponding server label. Therefore we can identify the idle periods in each server, and generate the corresponding unused potential departure events.

We still use Q (the number of customers in the system) as the indicator of all servers being busy ($Q \ge c$) or not ($Q < c$).

As for the arrivals, when $Q < c$, they are assigned to server $l = \min\{j : s^{(j)} = 0\}$ directly. Otherwise, they wait in the queue. The entrance into an idle server initiates a busy period in the corresponding server.

Algorithm 10 M/M/c FCFS queue backward simulation with m busy cycles

Initialize vectors INSTANTS and EVENTS to empty.
Simulate Q according to the p.m.f. of equation (4.17)
while $Q > 0$ **do**
  Simulate $X \sim \text{Exp}(\lambda_0 + (Q \wedge c)\mu_0)$
  $t \leftarrow t + X$
  Simulate E from $\{1, -1\}$ with $\Pr(E = 1) = \lambda_0/(\lambda_0 + (Q \wedge c)\mu_0)$
  Append t to INSTANTS, E to EVENTS and Q to QLENGTHS
  $Q \leftarrow (Q + E)^+$
end while
$M \leftarrow 0$ # counter of the busy cycles
while $M < m$ **do**
  Simulate $X \sim \text{Exp}(\lambda_0 + (Q \wedge c)\mu_0)$
  $t \leftarrow t + X$
  Simulate E from $\{1, -1\}$ with $\Pr(E = 1) = \lambda_0/(\lambda_0 + (Q \wedge c)\mu_0)$
  Append t to INSTANTS and E to EVENTS
  if $Q = 1$ and $E = -1$ **then**
    $M \leftarrow M + 1$
    $T \leftarrow t$
  end if
  $Q \leftarrow (Q + E)^+$
end while
Bind INSTANTS and EVENTS
Change the signs of EVENTS
Change the signs of INSTANTS and reverse the orders of INSTANTS and EVENTS
return $-T$, INSTANTS and EVENTS

For the departure, it occurs in server $l \sim \text{Unif}\{j : s^{(j)} = 1\}$, i.e. the server completing service at this instant is chosen randomly among all those busy, because the service durations share the same exponential distribution.

In server l, record the arrival instants which initiate a busy period, and the departure instants. These information are stored in fields of INSTANTS′ and SERVERS. An extra field of STATES is used to indicate the occupation states ("0" for being idle and "1" for being occupied) of the server found by the recorded events. E.g. a record of these three fields (INSTANTS′, SERVERS, STATES) is $(12.6, 1, 0)$. It indicates that an arrival event occurred at time 12.6. This arrival was assigned to server 1, and it found that this server was idle. Another example of record is $(24.7, 2, 1)$, which indicates a departure happened at time 24.7 in server 2.

As for server $l, l = 1, \ldots, c$, in each idle period (assuming its length is $\zeta \in \mathbb{R}$), which is identified by consecutive values of "1" and "0" stored in STATES, simulate $N \sim \text{Poi}(\zeta\bar{\mu})$, then generate N ordered uniform numbers on it as the unused potential departure instants.

The pseudocode of this algorithm is illustrated in Algorithm 11.

Note that if there are any idle servers at the two ends of interval $(\lfloor -T \rfloor, 0)$, we mark them with STATES = 1 at time $\lfloor -T \rfloor$, and STATES = 0 at time 0. They act as facilitating marks and will be deleted before outputting the results.

4.2.3 Upper bound of the carryover probability

In the coupled M/M/c FCFS queue, on interval $(k-1, k], k \in \mathbb{Z}$, the number of arrivals is N_k^A and that of potential departures N_k^D. Let Q_t^H be the number of customers in the M/M/c FCFS queue at time t. Block $\mathbb{C}_n$ and the individual job excess' upper bound have the same definitions as those in equations (4.8) and (4.9) respectively.

Proposition 4.2.2 *In the $M_t/M_t/c$ FCFS queue, the individual job excess' upper bound of block $\mathbb{C}_n$ after constructing the time-varying queue is less than the maximum number of arrivals in the unit intervals included in $\mathbb{C}_n$ plus $c-1$, i.e.*

$$\Omega_n \leq \mathbb{A}_n + c - 1,$$

where $\mathbb{A}_n = \max\left\{N_k^A, i = \lfloor \tau_n^{(0)} \rfloor + 1, \ldots, \lfloor \tau_n^{(1)} \rfloor\right\}$.

Proof Based on Proposition 4.2.1, we know that at the integral time points the dominating process of the $M_t/M_t/c$ FCFS queue is $L_k + c - 1, k \in \mathbb{Z}$. Since the dominating one is a single-server system, so the proof of Proposition 4.1.3 still holds in this scenario with the augmentation of constant $c-1$. □

Algorithm 11 Simulating the potential departure events in the $M/M/c$ FCFS queue

$Q \leftarrow Q^H_{\lfloor -T \rfloor}$
if $Q \geq c$ **then**
  Initialize $\vec{s}$ as $\vec{1}$ # $Q-$length vector of 1's
  Initialize INSTANTS′, SERVERS and STATES as empty
else
  Initialize $\vec{s}$ as $(1, \ldots, 1, 0, \ldots, 0)$ # The first Q elements are 1's, and the rest 0's.
  Initialize (INSTANTS′, SERVERS, STATES) with $c - Q$ records of $(\lfloor -T \rfloor, l, 1), l = Q + 1, \ldots, c$.
end if
for $i = 1$ to n **do**
  if EVENTS_i=1 **then**
    if $Q < c$ **then**
      $l \leftarrow \min\{j : s^{(j)} = 0\}$ # Choose the server for the arrival
      $s^{(l)} \leftarrow 1$ # Indicate it is occupied
      Append INSTANTS_i to INSTANTS′, l to SERVERS and 0 to STATES
    end if
    $Q \leftarrow Q + 1$
  else
    $l \sim \text{Unif}\{j : s^{(j)} = 1\}$ # Determine the server where the departure will occur
    Append INSTANTS_i to INSTANTS′, l to SERVERS and 1 to STATES
    if $Q \leq c$ **then**
      $s^{(l)} \leftarrow 0$
    end if
    $Q \leftarrow Q - 1$
  end if
end for
for $i \in \{j : s^{(j)} = 0\}$ **do**
  Append 0 to INSTANTS′, i to SERVERS and 0 to STATES # To mark the idle servers at time 0
end for
for $l = 1$ to c **do**
  Take the subgroup of INSTANTS′ and STATES with SERVERS = l
  for $i \in \{j : \text{STATES}_j = 1, \text{STATES}_{j+1} = 0\}$ **do**
    Simulate $N \sim \text{Poi}(\bar{\mu}(\text{INSTANTS}_{i+1} - \text{INSTANTS}_i))$
    Generate ordered uniform r.v.'s between INSTANTS_i and INSTANTS_{i+1}, then append them to INSTANTS′ with corresponding SERVERS = l and STATES = 1.
    Eliminate records in server l where STATES = 0
  end for
end for
Merge and save the potential departure events in all servers back into INSTANTS′ and SERVERS
Eliminate records of (INSTANTS′, SERVERS) with INSTANTS′ = $\lfloor -T \rfloor$ # Delete the marks
return INSTANTS′ and SERVERS

The bound of $\mathbb{A}_n$ (as defined in Proposition 4.2.2) has exactly the same form as shown in inequality (4.11), because the M/M/c FCFS queue is also a regenerative process and its arrival rate is denoted by $\bar{\lambda}$ too. The difference lies on the estimation of $\mathbb{E}(\mathcal{N})$. In the multi-server case, it has upper bound as $(1-\rho)^{-c}$ as shown in Proposition 3.5.4. Therefore we have

$$\Pr(\mathbb{A}_n > x) \leq m_n(1-\rho)^{-c}\frac{\overline{F}_{N^A}(x)}{1-e^{-\bar{\lambda}}}. \tag{4.18}$$

Let $N^A_{k,l}$ and $N^D_{k,l}, l \in \{1,\ldots,c\}$, be the numbers of arrival events and potential departure events respectively in server l on the interval $(k-1,k]$. It is clear that they are independent and

$$\begin{aligned} N^A_k &= \sum_{l=1}^{c} N^A_{k,l}, \qquad N^D_k = \sum_{l=1}^{c} N^D_{k,l}; \\ N^A_{k,l} &\sim \text{Poi}(\bar{\lambda}/c), \qquad N^D_{k,l} \sim \text{Poi}(\bar{\mu}). \end{aligned}$$

Since there are m_n totally idle periods in block $\mathbb{C}_n$, the extra capacity in it still has negative binomial distribution as we have shown in Proposition 4.1.2. Let $\mathcal{G}_{n,l}$ be the extra capacity in server $l, l \in \{1,\ldots,c\}$ for block $\mathbb{C}_n$. It is easy to see that

$$\mathcal{G}_{n,l} \sim \text{NB}(m_n, p), \tag{4.19}$$

where $p = \frac{\bar{\lambda}}{\bar{\lambda}+\bar{\mu}} = \frac{c\rho}{c\rho+1}$. The total extra capacity in block $\mathbb{C}_n$ is $\mathcal{G}_n = \sum_{l=1}^{c}\mathcal{G}_{n,l}$.

Now, we define $\mathcal{E}$ as the event that there is carryover, i.e. the job excess from all the blocks prior to $\lfloor\tau^{(0)}_{-1}\rfloor$ cannot be absorbed completely by the extra capacity in block $\mathbb{C}_{-1}$. Let $\mathcal{E}_{-j}, j = 1,2,\ldots$, be the event that the individual job excess from block $\mathbb{C}_{-(j+1)}$ cannot be absorbed completely by the extra capacity in block $\mathbb{C}_{-j}$. We do not reuse the definition about these concepts introduced in the single-server case (Section 4.1.4), because of the different behaviour of the multi-server queues. But, if none of events $\mathcal{E}_{-j}$ occurs, then $\mathcal{E}$ does not either. So the derivation is the same as that shown in Section 4.1.4, and we have

$$\Pr(\mathcal{E}) \leq \sum_{j=1}^{\infty}\Pr(\mathcal{E}_{-j}).$$

As for the upper bound of the individual job excess from block $\mathbb{C}_{-(j+1)}$ as show in Proposition 4.2.2, which is $\mathbb{A}_{-(j+1)} + c - 1$, it is hopefully absorbed by the extra capacity in block $\mathbb{C}_{-j}$. Denote by $\eta_l \geq 0, l \in \{1,\ldots,c\}$, the number of parts of the job excess consumed by server l in the successive block. So

$$\Pr(\mathcal{E}_{-j}) = \sum_{l=1}^{c}\Pr(\eta_l > \mathcal{G}_{-j,l}),$$

where $\sum_{l=1}^{c} \eta_l \leq \mathbb{A}_{-(j+1)} + c - 1$.

To reduce $\Pr(\mathcal{E}_{-j})$ to be less than a tiny number (ϵ), we generate multiple totally idle periods so that the mode of the distribution of $\mathcal{G}_{-j,l}$ (the extra capacity in each server) is very likely to be larger than the possible job excess. Therefore by considering the worst case where all of the job excess is allocated to only one server, we get the upper bound of

$$\begin{aligned} \Pr(\mathcal{E}_{-j}) &= \sum_{l=1}^{c} \Pr(\eta_l > \mathcal{G}_{-j,l}) \\ &\leq \Pr(\mathbb{A}_{-(j+1)} + c - 1 > \mathcal{G}_{-j,1}), \end{aligned} \tag{4.20}$$

where we choose server 1 since all servers are identical.

Similar to the derivation of inequality (4.14), based on inequalities (4.18) and (4.20), and the distribution specified in (4.19), we have

$$\Pr(\mathcal{E}_{-j}) < \frac{(1-\rho)^{-c} m_{-(j+1)}}{1 - e^{-\bar{\lambda}}} e^{\bar{\lambda}(e^t - 1) + t(c-1)} \left(\frac{c\rho}{c\rho + 1 - e^{-t}} \right)^{m_{-j}}. \tag{4.21}$$

When $c = 1$, it is exactly the corresponding inequality shown in the single-server case (equation 4.15).

Therefore the upper bound of the carryover probability becomes

$$\begin{aligned} \Pr(\mathcal{E}) &\leq \sum_{j=1}^{\infty} \Pr(\mathcal{E}_{-j}) \\ &\leq \frac{(1-\rho)^{-c} e^{\bar{\lambda}(e^t - 1) + t(c-1)}}{1 - e^{-\bar{\lambda}}} \sum_{j=1}^{\infty} m_{-(j+1)} \left(\frac{c\rho}{c\rho + 1 - e^{-t}} \right)^{m_{-j}}. \end{aligned}$$

Let $m_{-j} = j m_{-1}$, similar to equation (4.15), with specified ϵ and fixed t, the value of m_{-1} can be determined as

$$\begin{aligned} m_{-1}(t) = \min \Bigg\{ m : \frac{4(1-\rho)^{-c} e^{\bar{\lambda}(e^t - 1) + t(c-1)} m}{1 - e^{-\bar{\lambda}}} \left(\frac{c\rho}{c\rho + 1 - e^{-t}} \right)^{m} < \epsilon, \\ \left(\frac{c\rho}{c\rho + 1 - e^{-t}} \right)^{m} < \frac{1}{4}, m \in \mathbb{N} \Bigg\} \end{aligned} \tag{4.22}$$

In practice, we will choose t such that m_{-1} is an acceptable number.

4.2.4 Algorithm for nearly perfect sampling of the $M_t/M_t/c$ FCFS queue

The algorithm of nearly perfect sampling of the $M_t/M_t/c$ FCFS queue is similar to that of the $M_t/M_t/1$ case as shown in Section 4.1.4.

1. Compute m_{-1} according to equation (4.22).

2. Starting from time 0, simulate backwards an M/M/c FCFS queue (Algorithm 10) with arrival rate $\bar{\lambda}$ and service rate $\bar{\mu}$ for m_{-1} busy cycles, whose initial instant is $-T$. It is obvious that $-T = \tau_{-1}^{(0)}$, which is the beginning of block $\mathbb{C}_{-1}$. Record the arrival and departure instants in this procedure.

3. Continue simulating backwards the M/M/c FCFS queue to time $\lfloor -T \rfloor$. Keep recording the arrival and departure events and $Q^H_{\lfloor -T \rfloor}$.

4. Get the potential departure events according to Algorithm 11 with inputs of $Q^H_{\lfloor -T \rfloor}$ and the events generated in Steps 2 and 3.

5. On each unit interval from $\lfloor -T \rfloor$ to 0, according to equation (2.17) construct the coupled time-varying arrival instants with the homogeneous arrival instants generated in Steps 2 and 3. The time-varying potential departure instants (still bearing the association with server labels) are constructed from the homogeneous ones generated in Step 4.

6. Starting from $\lfloor -T \rfloor$ with $Q^N_{\lfloor -T \rfloor} = Q^H_{\lfloor -T \rfloor}$ and the coupled time-varying events, run the time-varying queue forward and output Q^N_0 as a stationary draw at time 0 of the $M_t/M_t/c$ FCFS queue.

Remark Proposition 3.5.1 supports that Q^N_0 is a steady-state draw at time 0 if there is no carryover.

4.3 Perfect sampling of $M_t/G/1$ queue

In this model, the arrival process is still a periodic Poisson process, but service durations (denoted by B) are homogeneous in time, and drawn from some general distribution ($G(\cdot)$). At an arrival instant, the service requirement of the customer can be simulated, thus we can follow the traditional way of analyzing the unfinished workload to explore this time-varying system. Since there is less uncertainty, i.e. the service duration distribution is not time dependent, it seems easier to handle compared with the $M_t/M_t/1$ queue.

As mentioned before, the first step is to find the dominating process. In the coming subsection we construct it as $\{V^H_k + 1\}_{k\in\mathbb{Z}}$, where V^H_k is the unfinished workload in the coupled homogeneous queue. It dominates the unfinished workload of the time-varying queue. Based on Proposition 3.5.1 we can get perfect sampling of the $M_t/G/1$ queue.

In the time-varying systems, the "unfinished workload" can be defined as the sum of the residual service durations of customers being presently served and the customers awaiting service. It comes from the definition of "workload" by Asmussen [5, p. 64]. Since there are jumps of the unfinished workload at the arrival instants, it is not continuous at these time points.

For two instants (denoted as Y_i and $Y_{i+1} \in \mathbb{R}$), which initiate two successive busy periods of a queueing system, we define the unfinished workload as

$$V_t = \sum_{j=1}^{N^A(Y_i, t-Y_i)} B_j - \int_0^{t-Y_i} (Q_s \wedge c) ds, \qquad t \in [Y_i, Y_{i+1}), i \in \mathbb{Z},$$

where $N^A(x, s)$, $s \geq 0$, is the number of arrivals on interval $[x, x+s]$, B_j $(j = 1, \ldots, N^A(x, s))$ the corresponding service durations, Q_t the number of customers in the system at time t, and $c \in \mathbb{N}$. In the single-server case, $c = 1$. With this definition, V_t is right continuous, i.e. $V_t = V_{t+}$. Note that $\{Y_i\}$ is no longer a renewal process.

4.3.1 The dominating process and its upper bound

Proposition 4.3.1 *Construct a coupled homogeneous queue $(M/G/1)$ by modifying a stable $M_t/G/1$ queue as follows. On each interval of $(k-1, k], k \in \mathbb{Z}$, let the number of arrivals in the $M_t/G/1$ queue be N_k^A. In the homogeneous queue let N_k^A customers arrive uniformly on this interval. Let the service durations in the homogeneous queue be the same values in the same order as those in the time-varying queue. Denote by V_k^H the unfinished workload at time k in the homogeneous queue, and by V_k^N that in the time-varying queue. Assume both of them are initially idle at time $t_0 \in \mathbb{Z}$, then*

$$V_k^N \leq V_k^H + 1, \ \forall k \in \mathbb{Z}, k \geq t_0.$$

Proof Clearly $V_k^N \leq V_k^H + 1$ for $k = t_0$, since both are 0. For larger k, let B_k be the additional workload that arrives during the interval (the same for both queues). Let W_k^N be the amount of work done on these new customers during the interval in the time-varying queue; W_k^H the counterpart in the homogeneous queue.

It is obvious that

$$\begin{aligned} V_k^N &= (V_{k-1}^N - 1)^+ + B_k - W_k^N, \\ V_k^H &= (V_{k-1}^H - 1)^+ + B_k - W_k^H. \end{aligned}$$

Note that $W_k^N \in [0,1)$, $W_k^N = 0$ if $V_{k-1}^N > 1$, and $V_k^N + W_k^N \leq 1$ if $V_{k-1}^N \leq 1$. Similar constraints hold for W_k^H.

So

$$V_k^N - V_k^H = (V_{k-1}^N - 1)^+ - (V_{k-1}^H - 1)^+ + W_k^H - W_k^N$$

We check all the possible cases:

1. If $V_{k-1}^N \leq 1$, then $V_k^N - V_k^H \leq W_k^H < 1$, which is our result.
2. Otherwise $V_{k-1}^N > 1$, then $W_k^N = 0$.

   (1) If $V_{k-1}^H \leq 1$, then

   $$\begin{aligned} &V_k^N - V_k^H = V_{k-1}^N - 1 + W_k^H \leq V_{k-1}^H + W_k^H \leq 1 \\ &\Rightarrow V_k^N \leq V_k^H + 1. \end{aligned}$$

   (2) If $V_{k-1}^H > 1$, then $W_k^H = 0$ and

   $$\begin{aligned} &V_k^N - V_k^H = V_{k-1}^N - V_{k-1}^H \leq 1 \\ &\Rightarrow V_k^N \leq V_k^H + 1. \end{aligned}$$

Based on the mathematical induction principle, our result holds. □

4.3.2 Algorithm for perfect sampling of $M_t/G/1$ FCFS queue

The perfect sampling of $M_t/G/1$ queue is performed by using the CFTP Block Absorption method (see Proposition 3.5.1). Since the coupled M/G/1 FCFS queue can be simulated backwards (see Algorithm 1, where the class numbers need not to be generated), this algorithm can be described as follows.

1. Starting from time 0, we simulate backwards the M/G/1 FCFS queue until it becomes idle for the first time, assuming at time $-T \in \mathbb{R}$.
2. Continue simulating backwards the M/G/1 FCFS queue until time $-T_1$, determined as follows. It should be the start of a busy cycle and the summation of the lengths of the idle periods exceeds 2 on the interval $(\lceil -T_1 \rceil, \lfloor -T \rfloor)$.

   Record the homogeneous arrival instants on $(\lceil -T_1 \rceil, 0)$ as INSTANTS, corresponding service durations as SERVICES, and unfinished workload at $\lceil -T_1 \rceil$ as $V^H_{\lceil -T_1 \rceil}$.

3. Generate the time-varying arrival instants by mapping the INSTANTS on interval $(\lceil -T_1 \rceil, 0)$ with equation (2.17). Denote the mapped instants as INSTANTS^N. The corresponding individual service durations remain the same.

4. Starting from $\lceil -T_1 \rceil$ with unfinished workload $V^N_{\lceil -T_1 \rceil} = V^H_{\lceil -T_1 \rceil}$, run the time-varying queue forward with INSTANTS^N and SERVICES, and output V^N_0 as a stationary draw of the unfinished workload at time 0 of the $M_t/G/1$ queue.

Proposition 4.3.2 *By following the algorithm for simulating the $M_t/G/1$ queue specified in Section 4.3.2, the output of V^N_0 is a stationary draw of the unfinished workload at time 0.*

Proof Assume an $M_t/G/1$ queue and an $M/G/1$ queue were started infinitely long ago and coupled in the way described in Proposition 4.3.1. So $V^N_k \leq V^H_k + 1, \forall k \in \mathbb{Z}$, hence $V^N_{\lceil -T_1 \rceil} \leq V^H_{\lceil -T_1 \rceil} + 1$.

Since V^H_t becomes zero on the interval $(\lfloor -T \rfloor, \lfloor -T \rfloor + 1)$, it must be that $V^H_{\lfloor -T \rfloor} < 1$. When mapping the homogeneous arrivals to be time-varying ones, the unfinished workload at time $\lfloor -T \rfloor$ being arranged into the interval prior to $\lfloor -T \rfloor$ must be less than 1. Because there are at least 2 units of idle time on the interval $(\lceil -T_1 \rceil, \lfloor -T \rfloor)$ in the homogeneous queue, there must be at least 1 unit of idle time in the coupled time-varying queue (starting at time $\lceil -T_1 \rceil$ with value $V^H_{\lceil -T_1 \rceil}$) on this interval. It can absorb the maximum difference (the extra single unit workload) completely.

Therefore conditions of Proposition 3.5.1 are fulfilled. Thus the result holds. □

4.3.3 Examples

Here we illustrate sampling the stationary unfinished workload in the $M_t/G/1$ FCFS queue with Erlang and Pareto distributions of the service durations. In both cases, the arrival rates are the same and they have the following form with periodic pattern.

$$\lambda(t) = 3 + 3\sin(2\pi t).$$

The service rates are $\mu = 4 = 1/\mathbb{E}(B)$.

- As for the Erlang case

  $$B \sim \Gamma(2, \theta),$$

  where $\Gamma(\alpha, \theta)$ stands for the standard Gamma distribution with shape parameter α and rate θ. Here we have

  $$\theta = \alpha\mu = 8.$$

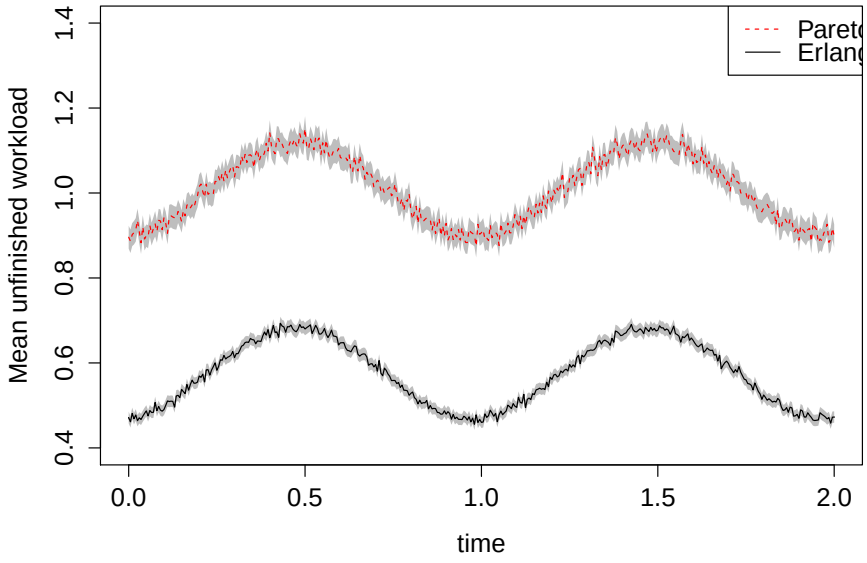

Figure 4.6: Average unfinished workloads and 95% confidence intervals (areas in gray shadow) in two $M_t/G/1$ FCFS queues with Erlang and Pareto distributions of the service durations. They involve 2 cycles and 100 points are drawn in each cycle. For each point we generate 10,000 samples.

- In the Pareto case the c.d.f. of service distribution has this form

$$G(x) = 1 - \left(\frac{\theta}{x+\theta}\right)^{\alpha}, x > 0.$$

  Assume

$$\alpha = 5 \Rightarrow \theta = \frac{\alpha - 1}{\mu} = 1.$$

The average unfinished workload and 95% confidence intervals are illustrated in Figure 4.6. It is clear that they behave in periodic patterns with the same periodic lengths as the arrival processes have.

4.4 Quick extensions to other models

Based on the algorithms developed before, it is easy to extend them to the queueing systems with multi-server or non-FCFS disciplines.

4.4.1 Perfect sampling of $\Sigma^K \mathrm{M}_t/\mathrm{G}/1$ or $\Sigma^K \mathrm{M}_t/\mathrm{G}_K/1$ APQ

Because the workload paths are invariant under any work-conserving disciplines in the single-server queueing systems, the perfect sampling algorithm of the $\mathrm{M}_t/\mathrm{G}/1$ FCFS queue developed in Section 4.3 leads to that of the $\Sigma^K \mathrm{M}_t/\mathrm{G}/1$ APQ in a straightforward fashion. Under the FCFS discipline, the $\Sigma^K \mathrm{M}_t/\mathrm{G}_K/1$ and $\mathrm{M}_t/\mathrm{G}/1$ queues are equivalent. Thus also leads to the solution of $\Sigma^K \mathrm{M}_t/\mathrm{G}_K/1$ APQ.

Assume τ is the most recent idle time in the $\mathrm{M}_t/\mathrm{G}/1$ FCFS queue, i.e. $\tau = \sup\{t : Q_t^N = 0, t \le 0, t \in \mathbb{R}\}$. Then by applying the APQ discipline to the arrival instants and associated service durations between τ and 0, starting from the empty state, we can get the steady-state draw of the coupled APQ at time 0.

4.4.2 Perfect sampling of $\mathrm{M}_t/\mathrm{G}/c$ FCFS queue

In the multi-server scenario, the FCFS allocation rule is the most efficient in the sense it has the smallest queue length. The "allocation rule" means the way to determine which server a customer should join in ([5, p. 341]). If the systems are initially empty, and they are fed with the same arrivals and the service durations which are used in the same order (see Lemma 1.3 by Asmussen [5, p. 342]), we can still use the RA model to dominate the $\mathrm{M}_t/\mathrm{G}/c$ FCFS queue, as we did in Section 3.4.

Since we can simulate the stationary idle time of the $\mathrm{M}_t/\mathrm{G}/1$ FCFS queue (Section 4.3), it is doable to find the most recent empty time of the $\mathrm{M}_t/\mathrm{G}/c$ RA model. Assume it is $\tau \in \mathbb{R}$. We sort the service durations according to their initiations in the RA model (see Section 3.4.2) and get a common sequence. Then we start from empty state at time τ, with the arrival instants simulated in the RA model, and the aligned service durations to construct the $\mathrm{M}_t/\mathrm{G}/c$ FCFS queue. Its state at time 0 is a steady-state draw.

4.4.3 Perfect sampling of $\Sigma^K \mathrm{M}_t/\mathrm{G}/c$ APQ

Under the common service distribution assumption, Proposition 3.4.1 can be extended to the time-varying case, because this dominance does not rely on the distribution of the inter-arrival times.

As described above, when restoring the $\mathrm{M}_t/\mathrm{G}/c$ FCFS queue forward, if we replace the FCFS with the APQ discipline, we get the steady-state draw of the $\Sigma^K \mathrm{M}_t/\mathrm{G}/c$ APQ at time 0.

Chapter 5

Conclusions and future work

Perfect sampling is an approach to directly sample from the steady state of an ergodic Markov chain without explicitly solving for it. In this thesis, we have achieved perfect samplings of a variety of non-preemptive work-conserving queues. Coupling From The Past (CFTP) and dominated CFTP were used in a variety of situations. Unlike the Regenerative Method (see Section 2.4.1), they both have finite expected run-time.

Coupling is the essential philosophy of the CFTP method. The key to dominated CFTP is to construct a dominating and reversible Markov chain. CFTP is quite appealing and practicable to treat queueing systems, because the stationary queues are deemed to become empty withing finite time, and many cases can be transformed into one-dimensional problems, where the workload or queue length are used to represent the system state. In the homogeneous scenarios, it is equivalent to find the regenerative time in the past. As for the time-varying systems, although the regenerative settings are restricted to certain epochs, the classical CFTP argument still supports the claim of steady-state draw at time 0, thus greatly simplified the algorithms.

5.1 Main contributions

The main contributions of this thesis are follows:

1. Nearly perfect sampling with well specified distance.

   In the homogeneous settings, as for the multi-server, multi-class and varying service distribution WCQs, since it hard to find the dominating process, we use the FCFS system to couple them. Because their workload paths are close, we can estimate the upper bound of the discrepancy and introduce extra blocks of busy cycles to absorb the workload excess. Thus leads to the so called CFTP "Block Absorption" method.

In the time-varying cases, such as $M_t/M_t/1$ FCFS queues, although we succeeded in constructing the dominating process, it hard to accomplish the backward simulation. Similarly, we use blocks of unused potential departure events to absorb the job excess.

The merit of this method lies in the well specified distance between the sample distribution and the target one. As shown in Proposition 3.5.5, the total variation is less than ϵ, which is the specified tolerance. To some extent, this method can be thought as an alternative way to get the transient sample. But since we start from a past time, it is easier to specify the distance more accurately compared to the ordinary simulation, which starts from an arbitrarily selected state and runs forward for a "burn-in" time.

2. Perfect and nearly perfect sampling of the time-varying queues with periodic Poisson arrival process.

   We design the coupling scheme by setting the numbers of arrivals the same on a complete cycle in the homogeneous and time-varying queues. The dominating processes are constructed by concentrating the arrivals of the complete cycle at the end of it.

   For the $M_t/M_t/1$ and $M_t/M_t/c$ FCFS queues, potential departure events are also concentrated at the beginning of these intervals in the dominating processes. Since it is hard to perform backward simulation of these dominating processes, the Regenerative Method is applied to achieve perfect sampling of the $M_t/M_t/1$ FCFS queue and CFTP Block Absorption for the nearly perfect sampling. As for the $M_t/M_t/c$ FCFS queue, we only implement nearly perfect sampling with CFTP Block Absorption, because the dominating process does not return to 0 thus we could not detect the regenerative point of the $M_t/M_t/c$ FCFS queue. Therefore we only obtain nearly perfect sampling of it with CFTP method.

   In the $M_t/G/1$ FCFS queues, compared to the coupled $M/G/1$ FCFS system, since the workload excess would not exceed the length of a complete cycle, and it can be absorbed by finite number of idle periods, this ensures that perfect sampling can be achieved. Furthermore, by using the RA model as the upper bound and keeping the service durations are used in the same order, we obtained perfect sampling of the $M_t/G/c$ FCFS queue.

   Because the workload paths are invariant for the $M_t/G/1$ FCFS and $\Sigma^K M_t/G/1$ WCQ, it is quite straightforward to extend the solution to the other work-conserving disciplines. Similarly, we can achieve the perfect sampling of the $\Sigma^K M_t/G/c$ WCQ, keeping in mind that the service durations should be used in the same order.

5.2 Future work

Since simulation based methods is a strong candidate for theoretically intractable problems, we can explore the following areas with perfect or nearly perfect sampling methods.

1. Queues with dependence.

   Firstly, we could explore the queueing system with Markovian Arrival Process (MAP) [44] [23, p. 98]. It models the dependence between successive inter-arrival times, while the service durations are usually i.i.d.'s and independent of the arrival process.

   Then we can extend the dependence to considering the autocorrelations of the inter-arrival times or service durations. They are likely to occur in the telecommunication networks or computer systems. Without counting these factors, models can predict overly optimistic performance measures. Livny et al. [39] presented some interesting results of the M/M/1 FCFS queue with these autocorrelations through a simulation study. Two methods (TES [42] and Minification / Maxification [38]) were introduced to generate the autocorrelated inter-arrival times and service durations, but the initialization bias was not considered, where the perfect sampling methods outperforms the ordinary simulation.

   Correlations between the inter-arrival time and service duration were also studied analytically, see [26] and [20]. This type of correlation can be applied to the ruin model, where the claim sizes and random incomes are correlated [62]. It is also a good chance to practice the perfect sampling methods, especially when the claim sizes (corresponding to the service durations) have heavy tail distributions.

2. Rare event simulation.

   Challenge arises when we are asked to figure out the probability of rare events (e.g. 10^{-9}), like the overflow of the buffer of some devices. The crude Monte Carlo does not work, because the relative error goes to infinity when the estimator of interest approaches zero [6, p. 158].

   Importance sampling is the commonly used method to deal with rare event simulation, see [7] and [24]. By tilting the original distribution, occurrences of the rare events are increased. Then we use a likelihood ratio to adjust the estimate under the new measure to recover the original one.

   A new method named "time reversal approach" to perform rare event simulation was proposed by Khanchi and Lamotheb [30]. Firstly, it estimates the "frequent event" with crude Monte Carlo method. Then it starts from the rare event to do simulation and uses the output to modify the probability estimated before.

Since we do not have the closed form of the target distribution, to combine the perfect sampling and the rare event simulation should be a non-trivial work.

Bibliography

[1] J. Abate and W. Whitt. Transient behavior of the $M/M/1$ queue: Starting at the origin. *Queueing Systems*, 2:41–65, 1987.

[2] J. Abate and W. Whitt. Transient behavior of the $M/G/1$ workload process. *Operations Research*, 42(4):750–764, 1994.

[3] J. Abate and W. Whitt. A unified framework for numerically inverting Laplace transforms. *Institute for Operations Research and the Management Sciences (INFORMS) Journal on Computing*, 18:408–421, 2006.

[4] A. S. Alfa and B. H. Margolius. Two classes of time-inhomogeneous markov chains: Analysis of the periodic case. *Annals of Operations Research*, 160(1):121–137, 2008.

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

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

[7] S. Asmussen and R. Y. Rubinstein. Steady state rare events simulation in queueing models and its complexity properties. In *in Advances in queueing*, pages 429–461. CRC Press, 1995.

[8] S. Asmussen and H. Thorisson. A markov chain approach to periodic queues. *Journal of Applied Probability*, 24(1):215–225, 1987.

[9] S. Asmussen, P. W. Glynn, and H. Thorisson. Stationarity detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation*, 2:130–157, 1992.

[10] O. J. Boxma. The longest service time in a busy period. *Mathematical Methods of Operations Research*, 24:235–242, 1980.

[11] Canadian Association of Emergency Physicians. The Canadian Triage and Acuity Scale. http://caep.ca/resources/ctas, 2013.

[12] M. L. Chaudhry and U. C. Gupta. Queue-length and waiting-time distributions of discrete-time $GI^X/Geom/1$ queueing systems with early and late arrivals. *Queueing Systems*, 25:307–324, 1997.

[13] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of scheduling*. Addison-Wesley Publishing Company, 1967.

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

[15] J. A. Fill. An interruptible algorithm for perfect sampling via Markov chains. *The Annals of Applied Probability*, 8:131–162, 1998.

[16] D. P. Gaver. Observing stochastic processes, and approximate transform inversion. *Operations Research*, 14(3):444–459, 1966.

[17] S. Ghahramani. Finiteness of moments of partial busy periods for $M/G/C$ queues. *Journal of Applied Probability*, 23(1):261–264, 1986.

[18] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.

[19] D. Gross and Carl M. Harris. *Fundamentals of Queueing Theory*. Wiley, 3rd edition, 1998.

[20] N. Hadidi. Further results on queues with partial correlation. *Operations Research*, 33 (1):203–209, 1985.

[21] J. M. Harrison and A. J. Lemoine. Limit theorems for periodic queues. *Journal of Applied Probability*, 14(3):566–576, 1977.

[22] A. M. Hasofer. On the single-server queue with non-homogeneous poisson input and general service time. *Journal of Applied Probability*, 1(2):369–384, 1964.

[23] Qi-Ming He. *Fundamentals of Matrix-Analytic Methods*. Springer New York, New York, 2014.

[24] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5(1):43–85, 1995.

[25] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[26] F. I. John. Single server queues with dependent service and inter-arrival times. *Journal of the Society for Industrial and Applied Mathematics*, 11(3):526–534, 1963.

[27] F. P. Kelly. *Reversibility and stochastic networks*. Chichester, 1979.

[28] D. G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 24 (3), 1953.

[29] W. S. Kendall and J. Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32(3):844–865, 2000.

[30] A. Khanchi and G. Lamotheb. Simulating tail asymptotics of a markov chain. *Statistics and Probability Letters*, 81(9):1392–1397, 2011.

[31] L. Kleinrock. A delay dependent queue discipline. *Naval Research Logistics Quarterly*, 11:329–341, 1964.

[32] L. Kleinrock. *Queueing Systems volume 1: Theory*. John Wiley & Sons, 1975.

[33] L. Kleinrock. *Queueing Systems volume 2: Computer Applications*. John Wiley & Sons, 1976.

[34] S. A. Klugman, H. H. Panjer, and G. E. Willmot. *Loss models : from data to decisions*. John Wiley & Sons, 3rd edition, 2008.

[35] J. Leal, F. Caballero, S. García-Sousa, P. Domingo, and A. López-Navidad. Distribution of organ donors on a weekly and annual basis according to cause of death. *Transplantation proceedings*, pages 2602–2603, 1999.

[36] A. J. Lemoine. Waiting time and workload in queues with periodic poisson input. *Journal of Applied Probability*, 26(2):390–397, 1989.

[37] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 1st edition, 2008.

[38] P. A. W. Lewis and E. McKenzie. Minification processes and their transformations. *Journal of Applied Probability*, 28(1):45–57, 1991.

[39] M. Livny, B. Melamed, and A. K. Tsiolis. The impact of autocorrelation on queuing systems. *Management Science*, 39(3):322–339, 1993.

[40] B. H. Margolius. A sample path analysis of the $M_t/M_t/c$ queue. *Queueing Systems*, 31: 59–93, 1999.

[41] B. H. Margolius. Transient and periodic solution to the time-inhomogeneous quasi-birth death process. *Queueing Systems*, 56:183–194, 2007.

[42] B. Melamed. Tes: A class of methods for generating autocorrelated uniform variates. *ORSA Journal on Computing*, 3(4):317–329, 1991.

[43] D. J. Murdoch and G. K. Takahara. Perfect sampling for queues and network models. *ACM TOMACS*, 16:76–92, 2006.

[44] M. F. Neuts. A versatile markovian point process. *Journal of Applied Probability*, 16(4): 764–779, 1979.

[45] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[46] S. B. Provost, M. Jiang, and H-T Ha. Moment-based approximations of probability mass functions with applications involving order statistics. *Communications in Statistics - Theory and Methods*, 38(12):1969To–1981, 2009.

[47] J.D. Rosendale and M.A. McBride. Organ donation in the united states: 1990-1998. *Clinical Transplants*, pages 83–94, 1999.

[48] S. M. Ross. *Stochastic Processes*. John Wiley & Sons, 2nd edition, 1996.

[49] S. M. Ross. *Introduction to Probability Models*. 10th edition, 2010.

[50] A. B. Sharif, D. A. Stanford, P. Taylor, and I. Ziedins. A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, 2014. doi: http://dx.doi.org/10.1016/j.orhc.2014.01.002.

[51] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue. *Journal of Applied Probability*, 48A:209–213, 2011.

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

[53] D. A. Stanford. Waiting and interdeparture times in priority queues with poisson- and general-arrival streams. *Operations Research*, 45(5):725–735, 1997.

[54] D. A. Stanford, P. Taylor, and I. Ziedins. Waiting time distributions in the accumulating priority queue. *Queueing Systems*, 77(3):297–330, 2014.

[55] H. Stehfest. Numerical inversion of Laplace transforms. *Communications of the ACM*, 13(1):47–49, 1970.

[56] D. P. Wiens. On the busy period distribution of the $M/G/2$ queueing system. *Journal of Applied Probability*, 26(4):pp. 858–865, 1989.

[57] D. B. Wilson. Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In Neal Madras, editor, *Monte Carlo Methods—Fields Institute Communications Vol. 26*. AMS, 2000.

[58] D. B. Wilson. How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms*, 16:85–113, 2000.

[59] R. W. Wolff. Upper bounds on work in system for multichannel queues. *Journal of Applied Probability*, 24(2):547–551, 1987.

[60] A. Zeifman, S. Leorato, E. Orsingher, Y. Satin, and G. Shilova. Some universal limits for nonhomogeneous birth and death processes. *Queueing Systems*, 52:139–151, 2006.

[61] J. Zhang. The transient solution of time-dependent M/M/1 queues. *IEEE Transactions on Information Theory*, 37:1690–1696, 1991.

[62] W. Zou, J. Gao, and J. Xie. On the expected discounted penalty function and optimal dividend strategy for a risk model with random incomes and interclaim-dependent claim sizes. *Journal of Computational and Applied Mathematics*, 255:270–281, 2014.

Curriculum Vitae

Name:	Yaofei Xiong
Post-Secondary Education and Degrees:	Beijing University of Posts and Telecommunications Beijing, China 1996 - 2000 B.Eng. Beijing University of Posts and Telecommunications Beijing, China 2000 - 2003 M.Eng. University of Western Ontario London, ON, Canada 2011 - 2012 M.Sc. University of Western Ontario London, ON, Canada 2012 - 2014 Ph.D.
Honours and Awards:	Excellent Graduate Student of Beijing City in 2000
Related Work Experience:	Teaching Assistant University of Western Ontario 2011 - 2014

Publications:

Perfect and Nearly Perfect Sampling of Work-Conserving Queues (submitted)