

Exact Simulation Techniques in Applied Probability and Stochastic Optimization

Yanan Pei

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2018

©2018

Yanan Pei

All Rights Reserved

ABSTRACT

Exact Simulation Techniques in Applied Probability and Stochastic Optimization

Yanan Pei

This dissertation contains two parts. The first part introduces the first class of perfect sampling algorithms for the steady-state distribution of multi-server queues in which the arrival process is a general renewal process and the service times are independent and identically distributed (iid); the *first-in-first-out* FIFO $GI/GI/c$ queue with $2 \leq c < \infty$. Two main simulation algorithms are given in this context, where both of them are built on the classical dominated coupling from the past (DCFTP) protocol. In particular, the first algorithm uses a coupled multi-server vacation system as the upper bound process and it manages to simulate the vacation system backward in time from stationarity at time zero. The second algorithm utilizes the DCFTP protocol as well as the Random Assignment (RA) service discipline. Both algorithms have finite expected termination time with mild moment assumptions on the interarrival time and service time distributions. Our methods are also extended to produce exact simulation algorithms for *Fork-Join* queues and infinite server systems.

The second part presents general principles for the design and analysis of unbiased Monte Carlo estimators in a wide range of settings. The estimators possess finite work-normalized variance under mild regularity conditions. We apply the estimators to various applications including unbiased steady-state simulation of regenerative processes, unbiased optimization in *Sample Average Approximations* and distribution quantile estimation.

Table of Contents

List of Figures	v
List of Tables	vi
Acknowledgements	vii
1 Introduction	1
I Exact Simulation of Multi-Dimensional Queueing Models with Renewal Input	3
2 Introduction to Part I	4
3 Exact Simulation with Vacation Systems	8
3.1 Simulation strategy and main result	8
3.1.1 Elements of the simulation strategy: upper bound and coupling	9
3.1.2 Monotonicity properties and the stationary $GI/GI/c$ queue	13
3.1.3 Description of simulation strategy and main result	17
3.2 Coalescence detection in finite time	18
3.3 Simulation procedure	23
3.3.1 Simulate a random walk with negative drift jointly with “milestone” events .	28
3.3.2 Simulate the vacation system between inspection times	30
3.3.3 Overall exact simulation procedure	31
3.4 Numerical experiments	32

4	Exact Simulation with Random Assignment	37
4.1	The FIFO and RA $GI/GI/c$ model	37
4.1.1	The FIFO $GI/GI/c$ model	37
4.1.2	The RA $GI/GI/c$ model	38
4.2	Simulating exactly from the stationary distribution of the RA $GI/GI/c$ model . . .	41
4.2.1	Algorithm for simulating exactly from π for the FIFO $GI/GI/c$ queue: The case $P(A > V) > 0$	45
4.2.2	Why we can assume that interarrival times are bounded	46
4.2.3	A more efficient algorithm: sandwiching	48
4.2.4	Continuous-time stationary constructions	51
4.3	Numerical experiments	53
4.4	Infinite server systems and other service disciplines	57
4.5	Fork-Join models	59
4.6	The case when $P(A > V) = 0$: Harris recurrent regeneration	60
II	Unbiased Monte Carlo Computations and Applications	62
5	Introduction to Part II	63
5.1	The general principles	65
6	Unbiased Multi-level Monte Carlo	70
6.1	Non-linear functions of expectations and applications	71
6.1.1	Application to steady-state regenerative simulation	75
6.1.2	Additional applications	76
6.2	Stochastic convex optimization	77
6.2.1	Unbiased estimator of optimal solution	80
6.2.2	Unbiased estimator of optimal value	84
6.2.3	Applications and numerical examples	86
6.3	Quantile estimation	89

III	Bibliography	92
	Bibliography	93
IV	Appendices	98
A	Appendix to Chapter 3	99
	A.1 The iid property of the coupled service times and independence of the arrival process	99
	A.2 Proof of technical lemmas of monotonicity	102
B	Appendix to Chapter 4	104
	B.1 Detailed algorithm steps in Section 4.2.1	104
	B.1.1 Simulation algorithm for the process $\{Y_{-n} : n \geq 0\}$	105
	B.1.2 Simulation algorithm for the process $\{X_{-n} : n \geq 0\}$	111
	B.1.3 Simulation algorithm for $\{S_n^{(r)} : n \geq 0\}$ and coalescence detection	115
	B.2 Proof of propositions	116

List of Figures

3.1	Renewal processes	11
3.2	The relationship between the renewal processes and the random walks (a) $N^0(t) - at$ and $S_n^{(0)}$ (b) $(a/c)t - N^i(t)$ and $S_n^{(i)}$	25
3.3	This figure plots a realization of the sample path $\{S_n^{(0)} : 0 \leq n \leq 11\}$. Here we set $m = 1$ and $L = 3$. Then, $\Phi_1^0 = 3$, $\Upsilon_1^0 = 4$, $\Phi_2^0 = 7$. If $\Upsilon_2^0 = \infty$, then for $n \geq 7$, $S_n^{(0)}$ will stay below the level $S_7^{(0)} + m$, which is demonstrated by the bold dashed line. Thus, $M_2^{(0)} = \max_{n \geq 2} \{S_n^{(0)}\} = S_2^{(0)}$ by only comparing the random walk values between step 2 and step 7.	27
3.4	Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 3$, $\mu = 2$ and $c = 2$	33
3.5	Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 10$, $\mu = 2$ and $c = 10$	34
3.6	Number of customers for an $Erlang(k_1, \lambda)/Erlang(k_2, \mu)/c$ queue in stationarity when $k_1 = 3$, $\lambda = 4.5$, $k_2 = 2$, $\mu = 2/3$, $c = 5$ and $\rho = 0.9$	34
4.1	Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 3$, $\mu = 2$, $c = 2$	54
4.2	Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 10$, $\mu = 2$, $c = 10$	54
4.3	Number of customers for an $Erlang(k_1, \lambda)/Erlang(k_2, \mu)/c$ queue in stationarity when $k_1 = 2$, $\lambda = 9$, $k_2 = 2$, $\mu = 5$, $c = 2$ and $\rho = 0.9$	55
4.4	Distributions of time taken to detect coalescence under two algorithms for an $M/M/c$ queue	56
6.1	Linear regression test on Beijing's PM2.5 data	87

6.2	Logistic regression test on AOL's campaign data	88
A.1	Matching procedure of service times to arrival process	100

List of Tables

3.1	Simulation estimates for the mean coalescence time of $M/M/c$ queue (QD)	35
3.2	Simulation estimates for the mean coalescence time of $M/M/c$ queue (QED)	35
3.3	Simulation result for computational complexities with varying traffic intensities	36
4.1	Simulation result for computational complexities with varying traffic intensities	57

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor Professor Jose Blanchet for his enthusiastic inspiration, patient guidance and continuous support through the five years of studies and research. Jose is a talented researcher, an insightful mentor and a caring friend. It has been an honor and a privilege working with Jose closely over my entire PhD years, and I enjoy every discussion with him as he always impress me with his ample knowledge, keen intuition and deep passion for research. The strong example he has set will continue to inspire me to stay curious, be passionate and strive towards excellence in my future work and life.

I am grateful to Professor Karl Sigman, Professor Ton Dieker, Professor Jing Dong and Professor Henry Lam for generously offering their time serving on my dissertation committee and providing invaluable advice. Special thanks go to Professor Jing Dong and Professor Karl Sigman, whom I collaborated along with Jose in the work of Chapter 3 and 4. I also want to thank Professor Xinyun Chen, with whom I worked on another research project about applying stochastic analysis techniques to uncover the structure of limit order books in financial markets. Although I finally decided not to include that work in the thesis because of the topic difference, I do appreciate this research experience, as it enlightened me on huge possibilities of connecting theories from academia to various practices in industry.

I would like to extend my sincere thanks to all the amazing professors who have taught me at Columbia University for sharing their knowledge without reservation, and all the members of staff at the department of IEOR for creating such a friendly and supportive environment.

Thanks to all my friends at Columbia, this journey has been colorful and enjoyable with your company. I will always cherish the memories of us attending the same courses, preparing qualification exams, discussing research ideas, exploring the great New York City, and sharing tears and joy together. My PhD life would be dull without you and I sincerely hope our friendship will last lifelong.

Lastly and most importantly, I owe my deepest gratitude towards my parents and grandmother, for their unconditional love and endless support throughout my life. They are my constant source of inspiration and strength. Special thanks to my husband for always believing in me and cheering me up. I would like to dedicate this work to them.

To Xiaoqin Liu, Yongqiang Pei, Luming Wang

Chapter 1

Introduction

This dissertation studies various bias-removal simulation techniques in applied probability and stochastic optimization problems. These techniques are developed based on a wide range of classic tools including queueing theory, steady-state analysis, perfect sampling, large deviation theory, multi-level Monte Carlo and sample average approximations. By different natures of such techniques, we divide the dissertation into two main parts.

Part I presents two sets of algorithms for simulating exactly from the stationary distribution of multi-server queues with general interarrival time and service time distributions in finite expected run time, and our work closes a gap in the perfect sampling literature. Perfect sampling aims to sample without any bias from the steady-state distribution of a given ergodic process, and it has evolved as a powerful way of sampling from stationary distributions of queueing models for which such distributions can not be derived explicitly. Both of the algorithms are developed by utilizing a perfect sampling protocol named Dominated Coupling From The Past (DCFTP), yet they are significantly different by design; one solves the problem by using a coupled multi-server vacation system as the upper bound process while the other directly simulates the Random Assignment (RA) model backward in time. We will have a thorough discussion about the background and relevant literatures of perfect sampling in Chapter 2, then we describe the two sets of algorithms respectively in Chapter 3 and Chapter 4.

Part II presents general de-biasing principles for Monte Carlo computations based on the multi-level Monte Carlo method and the bias removal ideas studied in the literature. Within the general framework, we propose unbiased estimators to various applied probability and operations research

settings such as steady-state simulation of regenerative processes, stochastic convex optimization, and distribution quantiles estimation. A key contribution of the development of such unbiased estimators is that it enables the use of parallel computing to improve the estimation accuracy and computation efficiency. In Chapter 5, we review the literatures and give the general principles to provide the high-level intuition. In Chapter 6, we discuss the construction of unbiased estimators for different settings of interest.

Part I

**Exact Simulation of
Multi-Dimensional Queueing Models
with Renewal Input**

Chapter 2

Introduction to Part I

In this part, we present two exact simulation algorithms for the steady-state distribution of multi-server queues with general interarrival time and service time distributions. Both of our algorithms have finite expected termination time under the assumption that the interarrival times and service times have finite $2 + \epsilon$ moments for some $\epsilon > 0$.

In recent years, the method of exact simulation has evolved as a powerful way of sampling from stationary distribution of a given ergodic process for which such distributions cannot be derived explicitly. The most popular perfect sampling protocol, known as Coupling From The Past (CFTP), was introduced in the seminal paper [Propp and Wilson, 1996]; see also [Asmussen *et al.*, 1992] for another important early reference on perfect simulation. Foss and Tweedie [Foss and Tweedie, 1998] proved that CFTP can be applied if and only if the underlying process is uniformly ergodic, which is not a property applicable to multi-server queues. So, we use a variation of the CFTP protocol called Dominated CFTP (DCFTP) introduced by Kendall in [Kendall, 1998] and later extended in [Kendall and Møller, 2000; Kendall, 2004].

A typical implementation of DCFTP requires at least four ingredients:

- (a) a stationary upper bound process for the target process,
- (b) a stationary lower bound process for the target process,
- (c) the ability to simulate (a) and (b) backward in time (i.e., from time 0 to $-t$, for any $t > 0$),
- (d) a finite time $-T < 0$ at which the state of the target process is determined (typically by

having the upper and lower bound processes coalesce), and the ability to reconstruct the target process from $-T$ up to time 0 coupled with the two bounding processes.

The time $-T$ is called the coalescence time, and it is desirable to have $E[T] < \infty$. The ingredients are typically combined as follows. One simulates (a) and (b) backward in time (by applying (c)) until the processes meet. The target process is sandwiched between (a) and (b). Therefore, if we can find a time $-T < 0$ when processes (a) and (b) coincide, the state of the target process is known at $-T$ as well. Then, applying (d), we reconstruct the target process from $-T$ up to time 0. The algorithm outputs the state of the target process at time 0.

It is quite intuitive that the output of the above construction is stationary. Specifically, assume that the sample path of the target process coupled with (a) and (b) is given from $(-\infty, 0]$. Then, we can think of the simulation procedure in (c) as simply observing or unveiling the paths of (a) and (b) during $[-t, 0]$. When we find a time $-T < 0$ at which the paths of (a) and (b) take the same value, because of the sandwiching property, the target process must share this common value at $-T$. Starting from that point, property (d) simply unveils the path of the target process. Since this path has been coming from the infinite distant past (we simply observed it from time $-T$), the output is stationary at time 0. Notice that while $-T$ is a random time, the output is the state of the target process at the fixed time 0.

One can often improve the performance of a DCFTP protocol if the underlying target process is monotone [Kendall, 2004], as in the multi-server queue setting. A process is monotone if there exists a certain partial order, \preceq , such that if w and w' are initial states where $w \preceq w'$, and one uses common random numbers to simulate two paths, one starting from w and the other from w' , then the order is preserved when comparing the states of these two paths at any point in time. Thus, instead of using the bounds (a) and (b) directly to detect coalescence, one could apply monotonicity to detect coalescence as follows: At any time $-t < 0$, one can start two paths of the target process, one from the state w' obtained from the upper bound (a) observed at time $-t$, and the other from the state $w \preceq w'$ obtained from the lower bound (b) observed at time $-t$. Then, we run these two paths using common random numbers, which are consistent with the backward simulation of (a) and (b), in reverse order according to the dynamics of the target process, and check whether these two paths meet before time zero. If they do, the coalescence occurs at such a meeting time. We also notice that because we are using common random numbers and system dynamics, these two

paths will merge into a single path from the coalescence time forward, and the state at time zero will be the desired stationary draw. If coalescence does not occur, then one can simply let $t \leftarrow 2t$, and repeat the above procedure. For this iterative search procedure, we must show that the search terminates in finite time.

While the DCFTP protocol is relatively easy to understand, its application is not straightforward. In most applications, the most difficult part has to do with element (c). Then, there is an issue of finding good bounding processes (elements (a) and (b)), in the sense of having short coalescence times – which we interpret as making sure that $E[T] < \infty$. There has been a substantial amount of research that develops generic algorithms for Markov chains (see, for example, [Corcoran and Tweedie, 2001] and [Connor and Kendall, 2007]). These methods rely on having access to the transition kernels, which are difficult to obtain in our case. Perfect simulation for queueing systems has also received a significant amount of attention in recent years, though most perfect simulation algorithms for queues impose Poisson assumptions on the arrival process. Sigman [Sigman, 2011; Sigman, 2012] applied the DCFTP and regenerative idea to develop perfect sampling algorithms for stable $M/G/c$ queues. The algorithm in [Sigman, 2011] requires the system to be super-stable (i.e., the system can be dominated by a stable $M/G/1$ queue). The algorithm in [Sigman, 2012] works under natural stability conditions by using a forward time regenerative method (a general method developed in [Asmussen *et al.*, 1992]) and using the $M/G/c$ model under a random assignment (RA) discipline as an upper bound, but it has infinite expected termination time. A recent work by Connor and Kendall [Connor and Kendall, 2015] extends Sigman’s algorithm [Sigman, 2012] by using the RA model. They accomplish this by first exactly simulating the RA model in stationary backward in time under process sharing (PS) at each node, then reconstructing it to obtain the RA model with FIFO at each node and doing so in such a way that a sample-path upper bound of the FIFO $M/G/c$ queue is achieved. Their algorithm has finite expected termination time, but it still requires the arrivals to be Poisson. The main reason for the Poisson arrival assumption is that under this assumption, one can find dominating processes which are quasi-reversible (see Chapter 3 of [Kelly, 1979]) and therefore can be simulated backward in time using standard Markov chain constructions (element (c)).

In general, constructing elements (a) and (b), (a) in particular, as (b) can often be taken as the trivial lower bound, $\mathbf{0}$, in the multi-server queue setting requires proving sample path (almost

sure) dominance under different service/routing disciplines. The sample path method has been widely used in the control of queues [Liu *et al.*, 1995]. Comparison of multi-server queues, under the almost sure dominance or the stochastic dominance, has been studied in the literature (see, for example, [Wolff, 1977; Foss, 1980; Foss and Chernova, 2001] and references therein).

For general renewal arrival process, our work is close in the spirit to [Ensor and Glynn, 2000], [Blanchet and Dong, 2013] and [Blanchet and Wallwater, 2015], but the models treated are fundamentally different. Thus, it requires some new developments.

For the first algorithm, we use a different coupling construction than that introduced in [Sigman, 2012] and refined in [Connor and Kendall, 2015]. In particular, we take advantage of a vacation system which allows us to transform the problem into simulating the running infinite horizon maximums (from time t to infinity) of renewal processes, compensated with negative drifts so that the infinite horizon maximums are well defined. Finally, we note that a significant advantage of our method, in contrast to [Sigman, 2012], is that we do not need to wait until the upper bound system empties to achieve coalescence. Due to the monotonicity of our process, we can apply the iterative method introduced above. This is important in many-server queues in heavy traffic for which it would take an exponential amount of time (in the arrival rate), or sometimes be impossible, to observe an empty system.

For the second set of algorithms, we utilize DCFTP by directly simulating the RA model in reverse-time (under FIFO at each node). The method involves extending, to a multi-dimensional setting, a recent result of [Blanchet and Wallwater, 2015] for exactly simulating the maximum of a negative drift random walk endowed with iid increments. An initial version of the algorithm is to simulate the upper bound process backward in time until an empty system is detected; then a more efficient “sandwiching” algorithm is given to deal with the cases where the empty status is difficult or impossible to observe. We also remark on how our approach can lead to new results for other models too, such as infinite server queues, multi-server queues under the *last-in-first-out* (LIFO) discipline, or the *randomly choose next* discipline, and even Fork-Join models (also called split and match models).

Chapter 3

Exact Simulation with Vacation Systems

We give our first exact simulation algorithm, which utilizes a so-called “vacation system” as an upper bound, in this chapter. In Section 3.1 we describe our simulation strategy, involving elements (a) – (d), and we conclude the section with the statement of a result which summarizes our main contribution (Theorem 1). Subsequent sections (Sections 3.2 and 3.3) provide more details of our simulation strategy. In Section 3.4, we conduct some numerical experiments. Appendix A contains the proofs of some technical results.

3.1 Simulation strategy and main result

Our target process is the stationary process generated by a multi-server queue with iid interarrival times and iid service times which are independent of the arrivals. There are $c \geq 1$ identical servers, each can serve at most one customer at a time. Customers are served on a *first-in-first-out* (FIFO) basis. Let $G(\cdot)$ and $\bar{G}(\cdot) = 1 - G(\cdot)$ (resp. $F(\cdot)$ and $\bar{F}(\cdot) = 1 - F(\cdot)$) denote the cumulative distribution function, CDF, and the tail CDF of the interarrival times (resp. service times). We shall use A to denote a random variable with CDF G , and V to denote a random variable with CDF F .

Assumption 1. *Both A and V are strictly positive with probability one, and there exists $\epsilon > 0$*

such that

$$E[A^{2+\epsilon}] < \infty, \quad E[V^{2+\epsilon}] < \infty.$$

The previous assumption will allow us to conclude that the coalescence time of our algorithm has finite expectation. The algorithm will terminate with probability one if $E[A^{1+\epsilon}] + E[V^{1+\epsilon}] < \infty$.

We assume that $G(\cdot)$ and $F(\cdot)$ are known so that the required parameters in our algorithmic development can be obtained. We write $\lambda = (\int_0^\infty \bar{G}(t)dt)^{-1} = 1/E[A]$ as the arrival rate, and $\mu = (\int_0^\infty \bar{F}(t)dt)^{-1} = 1/E[V]$ as the service rate. In order to ensure the existence of the stationary distribution of the system, we require the following stability condition: $\lambda/(c\mu) < 1$.

3.1.1 Elements of the simulation strategy: upper bound and coupling

We refer to the upper bound process as the *vacation system*, the construction that we use is based on that given in [Garmarnik and Goldberg, 2013]. Let us first explain in words how the vacation system operates. Customers arrive at the vacation system according to the renewal arrival process, and the system operates similarly to a $GI/GI/c$ queue, except that every time a server (say server i^*) finishes an activity (i.e., a service or a vacation), if there is no customer waiting to be served in the queue, server i^* takes a vacation which has the same distribution as the service times. If there is at least one customer waiting, the first customer waiting in the queue starts to be served by server i^* .

Using a suitable coupling, the work of [Garmarnik and Goldberg, 2013] shows that the total number of jobs in the vacation system is an upper bound of the total number of jobs in the corresponding multi-server queue. In this paper, we establish bounds for other system-related processes, such as the Kiefer-Wolfowitz vectors, which are of independent interest.

We next provide more details about the vacation system. We introduce $(c+1)$ time-stationary renewal processes, which are used to describe the vacation system.

Let

$$\mathcal{T}^0 := \{T_n^0 : n \in \mathbb{Z} \setminus \{0\}\}$$

be a time-stationary renewal point process with $T_n^0 > 0$ and $T_{-n}^0 < 0$, $n \geq 1$ (the T_n^0 are sorted in a non-decreasing order in n). For $n \geq 1$, T_n^0 represents the arrival time of the n -th customer into the system after time zero, and T_{-n}^0 is the arrival time of the n -th customer, counting backward in

time, from time zero. We also define

$$T_n^{0,+} := \inf\{T_m^0 : T_m^0 > T_n^0\},$$

that is, the arrival time of the next customer after T_n^0 . If $n \geq 1$ or $n \leq -2$, $T_n^{0,+} = T_{n+1}^0$. However, $T_{-1}^{0,+} = T_1^0$. Similarly, we write

$$T_n^{0,-} := \sup\{T_m^0 : T_m^0 < T_n^0\},$$

i.e., the arrival time of the previous customer before T_n^0 . Define $A_n := T_n^{0,+} - T_n^0$ for all $n \in \mathbb{Z} \setminus \{0\}$. Note that A_n is the interarrival time between the customer arriving at time T_n^0 and the next customer. A_n has CDF $G(\cdot)$ for $n \geq 1$ and $n \leq -2$, but A_{-1} has a different distribution due to the inspection paradox. Figure 3.1 (a) provides a pictorial illustration of the renewal process \mathcal{T}^0 .

Similarly, for $i \in \{1, 2, \dots, c\}$, we introduce iid time-stationary renewal point processes

$$\mathcal{T}^i := \{T_n^i : n \in \mathbb{Z} \setminus \{0\}\}.$$

As before, we have that $T_n^i > 0$ and $T_{-n}^i < 0$ for $n \geq 1$ with the T_n^i sorted in a non-decreasing order. We also define $T_n^{i,+} := \inf\{T_m^i : T_m^i > T_n^i\}$ and $T_n^{i,-} := \sup\{T_m^i : T_m^i < T_n^i\}$. Then, we let $V_n^i := T_n^{i,+} - T_n^i$. We assume that V_n^i has CDF $F(\cdot)$ for $n \geq 1$ and $n \leq -2$. The V_n^i are *activities* (services and vacations), which are executed by the i -th server in the vacation system.

Next, we define, for each $i \in \{0, 1, \dots, c\}$, and any $u \in (-\infty, \infty)$, a counting process

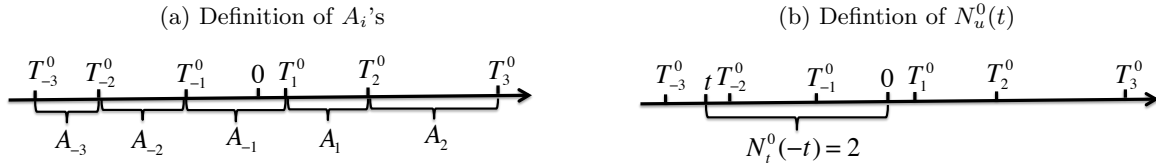
$$N_u^i(t) := |[u, u+t] \cap \mathcal{T}^i|,$$

for $t \geq 0$, where $|\cdot|$ denotes cardinality. Note that as $T_{-1}^i < 0 < T_1^i$ by stationarity, $N_0^i(0) = 0$. In particular, the quantity $N_u^0(t)$ is the number of customers who arrive during the time interval $[u, u+t]$ (see Figure 3.1 (b)). The quantity $N_u^i(t)$ is the number of activities initiated by server i during the time interval $[u, u+t]$ when $i \neq 0$. For simplicity of the notation, let us write $N^i(t) = N_0^i(t)$ if $t \geq 0$ and $N^i(t) = N_t^i(-t)$ if $t \leq 0$.

3.1.1.1 The upper bound process: vacation system

Let $Q_v(t)$ denote the number of people waiting in queue at time t in the stationary vacation system. We write $Q_v(t_-) := \lim_{s \uparrow t} Q_v(s)$ and $dQ_v(t) := Q_v(t) - Q_v(t_-)$. Also, for any $t \geq 0$, $i \in \{0, \dots, c\}$

Figure 3.1: Renewal processes



and each $u \in (-\infty, \infty)$, define

$$N_u^i(t_-) := \lim_{h \downarrow 0} N_{u-h}^i(t),$$

and let $dN_u^i(t) := N_u^i(t) - N_u^i(t_-)$ for all $t \geq 0$ (note that as $N_u^i(0_-) = 0$, $dN_u^i(0)$ should equal $N_u^i(0)$). Similarly, for $t \leq 0$, $N^i(t_-) = N_t^i(|t|_-)$.

We also introduce $X_u(t) := N_u^0(t) - \sum_{i=1}^c N_u^i(t)$. For simplicity of the notation, we also write $X(t) = X_0(t)$ if $t \geq 0$, and $X(t) = X_t(-t)$ if $t \leq 0$. Then, the dynamics of $(Q_v(t) : t > 0)$ satisfy

$$dQ_v(t) = dX(t) + I(Q_v(t_-) = 0) \sum_{i=1}^c dN^i(t), \quad (3.1)$$

given $Q_v(0)$. Note that here we are using the fact that arrivals do not occur at the same time as the start of activity times; this is because the processes \mathcal{T}^i are independent time-stationary renewal processes in continuous time so that T_{-1}^i and T_1^i have a density.

It follows from standard arguments of Skorokhod mapping [Chen and Yao, 2013] that, for $t \geq 0$,

$$Q_v(t) = Q_v(0) + X(t) - \inf_{0 \leq s \leq t} (X(s) + Q_v(0))^- ,$$

where $(X(s) + Q_v(0))^- = \min(X(s) + Q_v(0), 0)$. Moreover, using Loynes' construction, we have that, for $t \leq 0$,

$$Q_v(t) = \sup_{s \leq t} X(s) - X(t) \quad (3.2)$$

(see, for example, Proposition 1 of [Blanchet and Chen, 2015]). $(Q_v(t) : t \in (-\infty, \infty))$ is a well-defined process by virtue of the stability condition $\lambda/(\mu c) < 1$.

3.1.1.2 The coupling: extracting service times for each customer

The vacation system and the target process (the $GI/GI/c$ queue) will be coupled by using the same arrival stream of customers, \mathcal{T}^0 , and assuming that each customer brings his own service

time. In particular, the evolution of the underlying $GI/GI/c$ queue is described using a sequence of the form $((T_n^0, V_n) : n \in \mathbb{Z} \setminus \{0\})$, where V_n is the service time of the customer arriving at time T_n^0 . In simulation, we start by simulating the upper bound process (vacation system). Thus, the V_n must be extracted from the evolution of $Q_v(\cdot)$ so that the same service times are matched to the common arrival stream both in the vacation system and in the target process.

In order to match the service times to each of the arriving customers in the vacation system, we define the following auxiliary processes: For every $i \in \{1, \dots, c\}$, any $t > 0$, and any $u \in (-\infty, \infty)$, let $\sigma_u^i(t)$ denote the number of service initiations by server i during the time interval $[u, u + t]$. Observe that

$$\sigma_u^i(t) = \int_{[u, u+t]} I(Q_v(s_-) > 0) dN_u^i(s - u).$$

That is, we count activity initiations at time $T_k^i \in [u, u + t]$ as service initiations if and only if $Q_v(T_{k-}^i) > 0$. Once again, here we use the fact that arrival times and activity initiation times do not occur simultaneously.

We now explain how to match service time for the customer arriving at T_n^0 , $n \in \mathbb{Z} \setminus \{0\}$. First, such a customer occupies position $Q_v(T_n^0) \geq 1$ when he enters the queue. Let D_n^0 be the delay (or waiting time) inside the queue of the customer arriving at T_n^0 . Then we have that

$$D_n^0 = \inf \left\{ t \geq 0 : Q_v(T_n^0) = \sum_{i=1}^c \sigma_{T_n^0}^i(t) \right\},$$

and therefore,

$$V_n = \sum_{i=1}^c V_{N^i(T_n^0 + D_n^0)} \cdot dN^i(T_n^0 + D_n^0). \quad (3.3)$$

Observe that the previous equation is valid, because for each $n \in \mathbb{Z} \setminus \{0\}$, there is a unique $i(n) \in \{1, \dots, c\}$ for which $dN^{i(n)}(T_n^0 + D_n^0) = 1$ and $dN^j(T_n^0 + D_n^0) = 0$ if $j \neq i(n)$ (ties are not possible because of the time stationarity of the \mathcal{T}^i), so we obtain that (3.3) is equivalent to

$$V_n = V_{N^{i(n)}(T_n^0 + D_n^0)}.$$

We shall explain in Section A.1 that $(V_n : n \in \mathbb{Z} \setminus \{0\})$ and $(T_n^0 : n \in \mathbb{Z} \setminus \{0\})$ are two independent sequences and the V_n are iid copies of V , i.e., the extraction procedure here does not create any bias.

3.1.2 Monotonicity properties and the stationary $GI/GI/c$ queue

3.1.2.1 A family of $GI/GI/c$ queues and the target $GI/GI/c$ stationary system

We now describe the evolution of a family of standard $GI/GI/c$ queues. Once we have the sequence $((T_n^0, V_n) : n \in \mathbb{Z} \setminus \{0\})$, we can proceed to construct a family of continuous-time Markov processes $(Z_u(t; z) : t \geq 0)$ for each $u \in (-\infty, \infty)$, given the initial condition $Z_u(0; z) = z$. We write $z = (q, r, e(u))$, and set

$$Z_u(t; z) := (Q_u(t; z), R_u(t; z), E_u(t; z)),$$

for $t \geq 0$, where $Q_u(t; z)$ is the number of people in the queue at time $u+t$ ($Q_u(0; z) = q$), $R_u(t; z)$ is the vector of ordered (ascending) remaining service times of the c servers at time $u+t$ ($R_u(0; z) = r$), and $E_u(t; z)$ is the time elapsed since the previous arrival at time $u+t$ ($E_u(0; z) = e(u)$).

We shall always use $E_u(0; z) = e(u) = u - \sup\{T_n^0 : T_n^0 \leq u\}$, and we shall select q and r appropriately based on the upper bound. The evolution of the process $(Z_u(s; z) : 0 < s \leq t)$ is obtained by feeding the traffic $\{(T_n^0, V_n) : u < T_n^0 \leq u + s\}$ for $s \in (0, t]$ into a FIFO $GI/GI/c$ queue with initial conditions given by z . Constructing $(Z_u(s; z) : 0 < s \leq t)$ using the traffic trace $\{(T_n^0, V_n) : u < T_n^0 \leq u + s\}$ for $s \in (0, t]$ is standard (see, for example, Chapter 3 of [Rubinstein and Kroese, 2011]).

One can further describe the evolution of the underlying $GI/GI/c$ queue at arrival epochs, using the Kiefer-Wolfowitz vector [Asmussen, 2003]. In particular, for every non-negative vector $w \in \mathbb{R}^c$ such that $w^{(i)} \leq w^{(i+1)}$ (where $w^{(i)}$ is the i -th entry of w) for $1 \leq i \leq c-1$, and each $k \in \mathbb{Z} \setminus \{0\}$, the family of processes $\{W_k(T_n^0; w) : n \geq k, n \in \mathbb{Z} \setminus \{0\}\}$ satisfies

$$W_k(T_n^{0,+}; w) = \mathcal{S} \left((W_k(T_n^0; w) + V_n \mathbf{e}_1 - A_n \mathbf{1})^+ \right), \quad (3.4)$$

with initial condition $W_k(T_k^0; w) = w$, where $\mathbf{e}_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^c$, $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^c$, and \mathcal{S} is a sorting operator which arranges the entries in a vector in ascending order. In simple words, $W_k(T_n^0; w)$ for $k \in \mathbb{Z} \setminus \{0\}$ describes the Kiefer-Wolfowitz vector as observed by the customer arriving at T_n^0 , assuming that customer who arrived at T_k^0 , $k \leq n$, experienced the Kiefer-Wolfowitz state w .

Recall that the first entry of $W_k(T_n^0; w)$, namely $W_k^{(1)}(T_n^0; w)$, is the waiting time of the customer arriving at T_n^0 (given the initial condition w at T_k^0). More generally, the i -th entry of

$W_k(T_n^0; w)$, namely $W_k^{(i)}(T_n^0; w)$, is the virtual waiting time of the customer arriving at T_n^0 if he decides to enter service immediately after there are at least i servers free once he reaches the head of the line. In other words, one can also interpret $W_k(T_n^0; w)$ as the vector of remaining workloads (sorted in ascending order) that would be processed by each of the c servers at T_n^0 , if there are no more arrivals after time T_n^0 .

We are now ready to construct the stationary version of the $GI/GI/c$ queue. Namely, for each $n \in \mathbb{Z} \setminus \{0\}$ and every $t \in (-\infty, \infty)$, we define $W(n)$ and $Z(t)$ via

$$\begin{aligned} W(n) &:= \lim_{k \rightarrow -\infty} W_k(T_n^0; 0), \\ Z(t) &:= (Q(t), R(t), E(t)) = \lim_{u \rightarrow -\infty} Z_u(t - u, z_-(u)), \end{aligned} \tag{3.5}$$

where $z_-(u) = (0, 0, e(u))$. We shall show in Proposition 1 that these limits are well defined.

3.1.2.2 The analogue of the Kiefer-Wolfowitz process for the upper bound system

In order to complete the coupling strategy, we also describe the evolution of the analog Kiefer-Wolfowitz vector induced by the vacation system, which we denote by $(W_v(n) : n \in \mathbb{Z} \setminus \{0\})$, where v stands for vacation. As with the i -th entry of the Kiefer-Wolfowitz vector of a $GI/GI/c$ queue, the i -th entry of $W_v(n)$, namely $W_v^{(i)}(n)$, is the virtual waiting time of the customer arriving at time T_n^0 if he decides to enter service immediately after there are at least i servers free once he reaches the head of the line (assuming that servers become idle once they see, after the completion of current activity, the customer in queue waiting in the head of the line).

To describe the Kiefer-Wolfowitz vector induced by the vacation system precisely, let $U^i(t)$ be the time until the next renewal after time t in \mathcal{T}^i , that is $U^i(t) = \inf\{T_n^i : T_n^i > t\} - t$. So, for example, $U^0(T_n^0) = A_n$ for $n \in \mathbb{Z} \setminus \{0\}$. Let $U(t) = (U^1(t), \dots, U^c(t))^T$.

We then have that

$$W_v(n) = D_n^0 \mathbf{1} + \mathcal{S} \left(U \left((T_n^0 + D_n^0)_- \right) \right). \tag{3.6}$$

In particular, note that $W_v^{(1)}(n) = D_n^0$, i.e., the delay the customer arriving at T_n^0 would experience. We next introduce a recursive way of constructing/defining the Kiefer-Wolfowitz vector induced by the vacation system. We define

$$\bar{W}_v(n) = W_v(n) + V_n \mathbf{e}_1 - A_n \mathbf{1},$$

and let $\bar{W}_v^{(i)}(n)$ to be the i -th entry of $\bar{W}_v(n)$. Let $W_v(n+)$ denote the Kiefer-Wolfowitz vector seen by the customer arriving at $T_n^{0,+}$. From the definition of $W_v(n)$, we have

$$W_v(n+) = \mathcal{S} \left((\bar{W}_v(n))^+ + \Xi_n \right),$$

where

$$\Xi_n^{(i)} = I(\bar{W}_v^{(i)}(n) < 0) \cdot U^{j_i(n)} \left((T_n^{0,+})_- \right)$$

(i.e., $j_i(n)$ is the server whose remaining activity time immediately before T_n^0 is the i -th smallest in order).

So, (3.6) actually satisfies

$$W_v(n+) = \mathcal{S} \left((W_v(n) + V_n \mathbf{e}_1 - A_n \mathbf{1})^+ + \Xi_n \right), \quad (3.7)$$

where $\Xi_n = \left(\Xi_n^{(1)}, \dots, \Xi_n^{(c)} \right)^T$.

3.1.2.3 Monotonicity properties

In this section we will present several lemmas which contain useful monotonicity properties. The proofs of the lemmas are given in Section A.2 in order to quickly arrive at the main point of this section, which is the construction of a stationary version of the $GI/GI/c$ queue.

First, we recall that the Kiefer-Wolfowitz vector of a $GI/GI/c$ queue is monotone in the initial condition (3.8) and invoke a property (3.9) which will allow us to construct a stationary version of the Kiefer-Wolfowitz vector of our underlying $GI/GI/c$ queue, using Loynes' construction.

Lemma 1. For $n \geq k$, $k, n \in \mathbb{Z} \setminus \{0\}$, $w^+ > w^-$,

$$W_k(T_n^0; w^+) \geq W_k(T_n^0; w^-). \quad (3.8)$$

Moreover, if $k \leq k' \leq n$,

$$W_k(T_n^0; \mathbf{0}) \geq W_{k'}(T_n^0; \mathbf{0}). \quad (3.9)$$

The second result allows us to make precise how the vacation system dominates a suitable family of $GI/GI/c$ systems, in terms of the underlying Kiefer-Wolfowitz vectors.

Lemma 2. For $n \geq k$, $k, n \in \mathbb{Z} \setminus \{0\}$,

$$W_v(n) \geq W_k(T_n^0; W_v(k)).$$

The next result shows that in terms of queue length processes, the vacation system also dominates a family of $GI/GI/c$ queues, which we shall use to construct the upper bounds.

Lemma 3. *Let $q = Q_v(u)$, $r = \mathcal{S}(U(u))$, and $e = u - \sup\{T_n^0 : T_n^0 \leq u\}$, so that $z^+ = (q, r, e)$ and $z^- = (0, \mathbf{0}, e)$ then for $t \geq u$,*

$$Q_u(t - u; z^-) \leq Q_u(t - u; z^+) \leq Q_v(t).$$

Using Lemmas 1, 2, and 3, we can establish the following result.

Proposition 1. *The limits defining $W(n)$ and $Z(t)$ in (3.5) exist almost surely. Moreover, we have*

$$W_k(T_n^0; \mathbf{0}) \leq W(n) \leq W_k(T_n^0; W_v(k)). \quad (3.10)$$

Proof. Using Lemma 1 and 2, we have that

$$W_v(n) \geq W_k(T_n^0; W_v(k)) \geq W_k(T_n^0; \mathbf{0}).$$

Then, by property (3.9) in Lemma 1, we conclude that the limit defining $W(n)$ exists almost surely and that

$$W(n) \leq W_v(n). \quad (3.11)$$

Similarly, using Lemma 3, we can obtain the existence of the limit $Q(t)$ and we have that $Q(t) \leq Q_v(t)$. Moreover, by convergence of the Kiefer-Wolfowitz vectors, we obtain the i -th entry of $R(T_n^0 + W^{(1)}(n))$, namely

$$R^{(i)}(T_n^0 + W^{(1)}(n)) = \left(W^{(i)}(n) - W^{(1)}(n) \right)^+,$$

where $i \in \{1, \dots, c\}$. Lastly, since the age process has been taken from the underlying renewal process \mathcal{T}^0 , we have that $E(t) = t - \sup\{T_n^0 : T_n^0 \leq t\}$. The fact that the limits are stationary follows directly from the limiting procedure and it is standard in Loynes-type constructions.

For (3.10), we use the identity $W(n) = W_k(T_n^0; W(k))$, combined with Lemma 1, to obtain

$$W_k(T_n^0; \mathbf{0}) \leq W_k(T_n^0; W(k)) = W(n),$$

and then we apply Lemma 2, together with (3.11), to obtain

$$W(n) = W_k(T_n^0; W(k)) \leq W_k(T_n^0; W_v(k)).$$

□

3.1.3 Description of simulation strategy and main result

We now describe how the variation of DCFTP that we mentioned in Chapter 1, using monotonicity of the multi-server queue, and elements (a)–(d), apply to our setting.

Define a fixed inspection sequence $\{\kappa_j : j \geq 1\}$ with $\kappa_j < \kappa_{j-1} < 0$, and define $\kappa_0 = 0$. We start from the first inspection time $T_{\kappa_1}^0$ ($j = 1$). The upper bound is initialized using the Kiefer-Wolfowitz process associated with the vacation system at $T_{\kappa_j}^0$. The lower bound is initialized with a null vector $\mathbf{0}$. We run the two bounding $GI/GI/c$ queues forward in time using $\{(T_n^0, V_n) : \kappa_j \leq n \leq \kappa_{j-1}\}$. If the two processes meet before time zero, then we can “unveil” the state of the stationary $GI/GI/c$ queue; otherwise, we go backward in time to the next inspection time $T_{\kappa_{j+1}}^0$ ($j \leftarrow j+1$) and construct two new bounding $GI/GI/c$ queues accordingly. We repeat the procedure until the coalescence is detected.

The strategy combines the following facts (which we shall discuss in the sequel).

- *Fact I* We can simulate $\sup_{s \geq t} X(-s)$ and $(N^i(-t) : t \geq 0)_{i=0}^c$ jointly for any given $t \geq 0$. This part, which corresponds to item (c), is executed by applying an algorithm from [Blanchet and Wallwater, 2015] designed to sample the infinite horizon running time maximum of a random walk with negative drift. We shall provide more details about this in Section 3.3.
- *Fact II* For all $k \leq -1$ and every $k \leq n \leq -1$, by Proposition 1, we have that

$$W_k(T_n^0; \mathbf{0}) \leq W(n) \leq W_k(T_n^0; W_v(k)).$$

This portion exploits the upper bound (a) (i.e., $W_v(k)$) and the lower bound (b) (i.e., $\mathbf{0}$).

- *Fact III* We can detect that coalescence occurs at some time $T \in [T_k^0, 0]$ for some $k \leq -1$ by finding some $n \in \mathbb{Z}_-$, $n \geq k$, such that $T_n^0 + W_k^{(1)}(T_n^0; W_v(k)) \leq 0$ and

$$W_k(T_n^0; W_v(k)) = W_k(T_n^0; \mathbf{0}).$$

This is precisely the coalescence detection strategy which uses monotonicity of the Kiefer-Wolfowitz vector and the coalescence time $T = T_n^0 + W_k^{(1)}(T_n^0; W_v(k))$.

- *Fact IV* We can combine Facts I-III to conclude that

$$Z_{T_k^0}^0(|T_k^0|; Q(T_k^0), \mathcal{S}(U(T_k^0)), 0) = Z(0) \quad (3.12)$$

is stationary. We also have that

$$W_k(T_1^0; 0) = W(1),$$

which follows the stationary distribution of the Kiefer-Wolfowitz vector of a $GI/GI/c$ queue.

The main result of this paper is stated in the following theorem.

Theorem 1. *If Assumption 1 is in force, with $\lambda/(c\mu) \in (0, 1)$. Then, Facts I–IV hold true. We can detect coalescence at a time $-T < 0$ such that $E[T] < \infty$.*

The rest of the chapter is dedicated to the proof of Theorem 1. We have verified a number of monotonicity properties in Section 3.1.2.3, which in particular allow us to conclude that the construction of $W(n)$ and $Z(t)$ is legitimate (i.e., the limits exist almost surely). The monotonicity properties also yield Fact II and pave the way to verify Fact III. Section 3.2 proves the finite expectation of the coalescence time. In Section 3.3, we provide more algorithmic details about our perfect sampling construction.

3.2 Coalescence detection in finite time

In this section, we give more details about the coalescence detection scheme. The next result corresponds to Fact III and Fact IV.

Proposition 2. *Suppose that $w^+ = W_v(k)$ for some $k \leq -1$, and $w^- = \mathbf{0}$. If $W_k(T_n^0; w^+) = W_k(T_n^0; w^-)$ for some $k \leq n \leq -1$, then $W_k(T_m^0; w^+) = W(m) = W_k(T_m^0; w^-)$ for all $m \geq n$. Moreover, for all $t \geq T_n^0 + W_k^{(1)}(T_n^0; w^+)$,*

$$Z_{T_k^0}(t - T_k^0; (Q_v(T_k^0), \mathcal{S}(U(T_k^0)), 0)) = Z_{T_k^0}(t - T_k^0; (0, \mathbf{0}, 0)) = Z(t). \quad (3.13)$$

Proof. The fact that

$$W_k(T_m^0; w^+) = W(m) = W_k(T_m^0; w^-)$$

for $m \geq n$ follows immediately from the recursion defining the Kiefer-Wolfowitz vector. Now, to show the first equality in (3.13), it suffices to consider $t = T_n^0 + W_k^{(1)}(T_n^0; w^+)$, since from $t \geq T_n^0$ the input is exactly the same and everyone coming after T_n^0 will depart the queue and enter service

after time $T_n^0 + W_k^{(1)}(T_n^0; w^+)$. The arrival processes (i.e., $E_u(\cdot)$) clearly agree, so we just need to verify that the queue lengths and the residual service times agree. First, note that

$$\begin{aligned}
& R_{T_k^0}(T_n^0 + W_k^{(1)}(T_n^0; w^+) - T_k^0; (Q_v(T_k^0), \mathcal{S}(U(T_k^0)), 0)) \\
&= W_k(T_n^0; w^+) - W_k^{(1)}(T_n^0; w^+) \cdot \mathbf{1} \\
&= W_k(T_n^0; w^-) - W_k^{(1)}(T_n^0; w^-) \cdot \mathbf{1} \\
&= R_{T_k^0}(T_n^0 + W_k^{(1)}(T_n^0; w^-) - T_k^0; (0, \mathbf{0}, 0)).
\end{aligned} \tag{3.14}$$

So, the residual service times of both upper and lower bound processes agree. The agreement of the queue lengths follows from Lemma 3. Finally, the second equality in (3.13) follows from Proposition 1. \square

Next, we analyze properties of the coalescence time. Define

$$\begin{aligned}
T_- = \sup\{T_k^0 \leq 0 : \inf_{T_k^0 \leq t \leq 0} \| & Z_{T_k^0}(t - T_k^0; (Q_v(T_k^0), \mathcal{S}(U(T_k^0)), 0)) \\
& - Z_{T_k^0}(t - T_k^0; (0, \mathbf{0}, 0))\|_\infty = 0\}.
\end{aligned}$$

Notice that if at time T_- we start an upper bound queue,

$$Z_{T_-}(\cdot; (Q_v(T_-), \mathcal{S}(U(T_-)), 0)),$$

and a lower bound queue, $Z_{T_-}(\cdot; (0, \mathbf{0}, 0))$, they will coalesce before time 0. Thus, if we simulate the system up to T_- , we will be able to detect a coalescence. We next establish that $E[|T_-|] < \infty$.

By stationarity, we have that $|T_-|$ is equal in distribution to

$$T = \inf \left\{ T_k^0 \geq 0 : \inf_{0 \leq t \leq T_k^0} \| Z_0(t; (Q_v(0), \mathcal{S}(U(0)), 0)) - Z_0(t; (0, \mathbf{0}, 0)) \|_\infty = 0 \right\}.$$

Proposition 3. *If $E[V_n] < cE[A_n]$ for $n \geq 1$ and Assumption 1 holds,*

$$E[T] < \infty.$$

Proof. Define

$$\tau = \inf \{ n \geq 1 : W_1(T_n^0; W_v(1)) = W_1(T_n^0; 0) \}.$$

By Wald's identity, $E[A_n] < \infty$, for any $n \geq 1$; it suffices to show that $E[\tau] < \infty$.

We start with an outline of the proof, which involves two main components. I) We first construct a sequence of events which lead to the occurrence of τ . The events that we construct put constraints on the interarrival times and service times so that we see a decreasing trend on the Kiefer-Wolfowitz vectors. When putting a number of these events together (consecutively), the waiting time of the upper bound system will drop to zero. We further impose the events for c more arrivals after the waiting time drops to zero. Notice that these c arrivals do not have to wait in both the upper bound and the lower bound systems. Thus, by the time of c -th such arrival, the two systems will have the same set of customers with the same remaining service times. II) Based on events constructed in I, we then split the process $\{W_1(T_n^0; W_v(1)) : n \geq 1\}$ into cycles where: IIa) the probability that the desired event, which leads to coalescence, happens during each cycle is bounded from below by a positive constant, and IIb) the expected cycle length is bounded from above by a constant. IIa allows us to bound the number of cycles we need to check before finding τ by a geometric random variable. Then, we apply Wald's identity using IIb to establish an upper bound for $E[\tau]$.

We next provide more details of the proof, which are divided into part I and II as outlined above.

Part I We first construct the sequence of events, $\{\Omega_k : k \geq 2\}$, which enjoys the property that if Ω_k happens, the two bounding systems would have coalesced by time of the $(k + \lceil cK/\epsilon \rceil - 1)$ -th arrival.

As $E[V_n] < cE[A_n]$, for $n \geq 2$, we can find $m, \epsilon > 0$ such that for every $n \geq 2$, the event $H_n = \{V_n < cm - \epsilon, A_n > m\}$ is nontrivial in the sense that $P(H_n) > \delta$ for some $\delta > 0$. Now, pick $K > cm$ large enough, and define, for $k \geq 2$,

$$\Omega_k = \left\{ W_1^{(c)}(T_k^0; W_v(1)) \leq K \right\} \bigcap_{n=k}^{k+\lceil cK/\epsilon \rceil - 1} H_n.$$

To see the coalescence of the two bounding systems, let $\tilde{W}_k = (K, K, \dots, K)^T$ be a c -dimensional vector with each element equal to K . We notice that, under Ω_k ,

$$\tilde{W}_k \geq W_1(T_k^0; W_v(1)).$$

For $n \geq k$, define $\tilde{V}_n = cm - \epsilon$, $\tilde{A}_n = m$, and the (auxiliary) Kiefer-Wolfowitz sequence

$$\tilde{W}_{n+1} = \mathcal{S} \left(\left(\tilde{W}_n + \tilde{V}_n \mathbf{e}_1 - \tilde{A}_n \mathbf{1} \right)^+ \right).$$

Then, Ω_k implies $V_n < \tilde{V}_n$ and $A_n > \tilde{A}_n$ for $n \geq k$, which in turn implies

$$W_1(T_n^0; W_v(1)) \leq \tilde{W}_n.$$

Moreover, under Ω_k , we have

$$\tilde{W}_n^{(1)} = 0 \text{ and } \tilde{W}_n^{(c)} < cm \text{ for } n = k + \lceil cK/\epsilon \rceil - c + 1, \dots, k + \lceil cK/\epsilon \rceil.$$

Then, $W_1^{(1)}(T_n^0; W_v(1)) = 0$ and $W_1^{(c)}(T_n^0; W_v(1)) < cm$ for $n = k + \lceil cK/\epsilon \rceil - c + 1, \dots, k + \lceil cK/\epsilon \rceil$.

This indicates that under Ω_k , (1) all the arrivals between the $(k + \lceil cK/\epsilon \rceil - c + 1)$ -th arrival and the $(k + \lceil cK/\epsilon \rceil)$ -th arrival (included) enter service immediately upon arrival (have zero waiting time), and (2) the customers initially seen by the $(k + \lceil cK/\epsilon \rceil - c + 1)$ -th arrival would have left the system by the time of the $(k + \lceil cK/\epsilon \rceil)$ -th arrival. The same analysis holds assuming that we replace $W_1(T_k^0; W_v(1))$ by $W_1(T_k^0; 0)$. Therefore, by the time of the $(k + \lceil cK/\epsilon \rceil - 1)$ -th arrival, the two bounding systems would have exactly the same set of customers with exactly the same remaining service times, which is equal to their service times minus the time elapsed since their arrival times (since all of them start service immediately upon arrival). We also notice that since there is no customer waiting, the sorted remaining service time at $T_{k+\lceil cK/\epsilon \rceil-1}^0$ coincides with the Kiefer-Wolfowitz vector $W_{k+\lceil cK/\epsilon \rceil-1}$.

Part II We first introduce how to split the process into cycles, which are denoted as $\{(\tilde{\kappa}_i, \tilde{\kappa}_{i+1}), i \geq 1\}$. Let $\mathcal{U}_K := \{w : w^{(c)} \leq K\}$. We define

$$\tilde{\kappa}_1 := \inf\{n \geq 1 : W_1(T_n^0; W_v(1)) \in \mathcal{U}_K\},$$

and for $i \geq 2$, define

$$\tilde{\kappa}_i := \{n > \tilde{\kappa}_{i-1} + \lceil cK/\epsilon \rceil - 1 : W_1(T_n^0; W_v(1)) \in \mathcal{U}_K\}.$$

We denote $\Theta_i = \bigcap_{n=\tilde{\kappa}_i}^{\tilde{\kappa}_i + \lceil cK/\epsilon \rceil - 1} H_n$ for $i \geq 1$. We next show that the event Θ_i happens during the i -th cycle with positive probability. Since $P(H_n) > \delta$, $P(\Theta_i) \geq \delta^{\lceil cK/\epsilon \rceil} > 0$. Let N denote the first i for which Θ_i occurs. Then, N is stochastically bounded by a geometric random variable with probability of success $\delta^{\lceil cK/\epsilon \rceil}$. In particular, $E[N] \leq \delta^{-\lceil cK/\epsilon \rceil} < \infty$.

We next show that $E[\tilde{\kappa}_{i+1} - \tilde{\kappa}_i]$ is bounded using the standard Lyapunov argument. Under Assumption 1 and $\lambda < c\mu$, $\{W_1(T_n^0; w(1)) : n \geq 1\}$ for any fixed initial condition $w(1)$ is a positive

recurrent Harris chain [Asmussen, 2003]. Under Assumption 1, we also have that $(Q_v(t) : t \in (-\infty, \infty))$ is a well-defined process with $E[Q_v(t)] < \infty$ (see the random-walk bound in (3.18)).

Thus,

$$E \left[\sum_{i=1}^c W_v^{(i)}(1) \right] < \infty.$$

Consider the Lyapunov function $g(W) = W^{(c)}$, i.e., $g(W) \geq 0$ and $g(W) \rightarrow \infty$ as $\|W\| \rightarrow \infty$.

Then, for K large enough, as $\lambda < c\mu$, we can find $\delta \in (0, c/\lambda - 1/\mu)$ such that

$$E [g(W_1(T_{c+1}^0, w(1)))] \leq g(w(1)) - \delta \text{ for } w(1) \notin \mathcal{U}_K. \quad (3.15)$$

We also have

$$E [g(W_1(T_{c+1}^0, w(1)))] \leq K + c/\mu \text{ for } w(1) \in \mathcal{U}_K.$$

Then, by Theorem 2 in [Foss and Konstantopoulos, 2006], $E[\tilde{\kappa}_1] < \infty$ and we can find a constant $M > 0$ such that $E[\tilde{\kappa}_i - \tilde{\kappa}_{i-1}] < M$ for $i \geq 2$. We comment that here we need to look c steps ahead to identify the downward drift in (3.15), Thus, we use a general version of Lyapunov argument developed in [Foss and Konstantopoulos, 2006].

Lastly, by Wald's identity we have (setting $\tilde{\kappa}_0 = 0$) that

$$\begin{aligned} E[\tau] &\leq E[\tilde{\kappa}_N] + \lceil cK/\epsilon \rceil - 1 \\ &= E \sum_{i=1}^N (\tilde{\kappa}_i - \tilde{\kappa}_{i-1}) + \lceil cK/\epsilon \rceil - 1 \\ &\leq E[N] \times M + E[\tilde{\kappa}_1] + \lceil cK/\epsilon \rceil - 1 < \infty. \end{aligned}$$

□

Remark 1. *Following the proof, we can also conclude that the number of “activities” (either vacations or services) to simulate in the vacation system, denoted as N_V , is also finite in expectation. Since coalescence is detected by the τ -th arrival, we only need to simulate the vacation system forward in time from time 0 until we are able to extract the first $Q_v(0) + \tau$ service time requirements to match the customers waiting in queue at time 0 and the arrivals from time 0 to coalescence time T .*

For any $m' < \infty$ such that $E[V \wedge m'] > 0$, we let $\bar{N}^i(t)$, $i = 1, \dots, c$, denote the counting process corresponding to the i -th “truncated” vacation process with independent activity times capped by

m' , i.e., $V \wedge m'$. Following a standard argument as in the proof of Ward's identity in [Asmussen, 2003], a loose upper bound for $E[N_v]$ is given by

$$\begin{aligned} E[N_V] &\leq \sum_{i=1}^c E[N^i(T) + 1] + E[Q_v(0) + \tau] \\ &\leq \sum_{i=1}^c E[\bar{N}^i(T) + 1] + E[Q_v(0)] + E[\tau] \\ &\leq c \cdot \frac{E[T] + m'}{E[V \wedge m']} + E[Q_v(0)] + E[\tau] < \infty. \end{aligned}$$

3.3 Simulation procedure

In this section, we first address the validity of Fact I, namely, that we can simulate the vacation system backward in time, jointly with $\{T_n^i : m \leq n \leq -1\}$ for $0 \leq i \leq c$, for any $m \in \mathbb{Z}_-$.

Let $G_e(\cdot) = \lambda \int_0^\infty \bar{G}(x) dx$ and $F_e(\cdot) = \mu \int_0^\infty \bar{F}(x) dx$ denote equilibrium CDFs of the interarrival time and service time distributions, respectively. We first notice that simulating the stationary arrival process $\{T_n^0 : n \leq -1\}$ and stationary service/vacation completion process $\{T_n^i : n \leq -1\}$ for each $1 \leq i \leq c$ is straightforward by the reversibility of \mathcal{T}_n^i for $0 \leq i \leq c$. Specifically, we can simulate the renewal arrival process forward in time from time 0 with the first interarrival time following G_e and subsequent interarrival times following G . We then set $T_{-k}^0 = -T_k^0$ for all $k \geq 1$. Likewise, we can also simulate the service/vacation process of server i , for $i = 1, \dots, c$, forward in time from time 0 with the first service/vacation initiation time following F_e and subsequent service/vacation time requirements distributed as F . Let $T_k^i, k \geq 1$, denote the k -th service/vacation initiation time of server i counting forward in time. Then, we set $T_{-k}^i = -T_k^i$.

Similarly, we have the equality in distribution, for all $t \geq 0$ (jointly),

$$X(-t) \stackrel{d}{=} X(t);$$

therefore, we have from (3.2) that the following equality in distribution holds for all $t \geq 0$ (jointly):

$$Q_v(-t) \stackrel{d}{=} \sup_{s \geq t} X(s) - X(t).$$

The challenge in simulating $Q_v(-t)$ involves sampling $M(t) = \max_{s \geq t} \{X(s)\}$ jointly with $X(t)$ during any time interval of the form $[0, T]$ for $T > 0$. The rest of the section is devoted to solve this challenge.

The idea is to identify a sequence of random times Δ_k such that

$$\max_{T_k^0 \leq t \leq T_k^0 + \Delta_k} \{X(t)\} \geq \max_{t \geq T_k^0 + \Delta_k} \{X(t)\}.$$

Then, $M(T_k^0) = \max_{t \geq T_k^0} \{X(t)\} = \max_{T_k^0 \leq t \leq T_k^0 + \Delta_k} \{X(t)\}$. In particular, to calculate $M(T_k^0)$, we only need to look at the maximum of $X(t)$ over a *finite* time interval, $[T_k^0, T_k^0 + \Delta_k]$. To find such Δ_k , we apply two tricks here. The first trick is to decompose $X(t)$ into $(c + 1)$ random walks with negative drift associated with N^i for $i = 0, 1, \dots, c$. This is based on the fact that for $\lambda < c\mu$, we can pick $a \in (\lambda, c\mu)$, such that $N^0(t) - at$ and $((a/c)t - N^i(t))$ are “drifted downward” to negative infinity. We can then bound $M(t)$ by the “corresponding” running time maximum of the random walks with negative drift. The second trick is a “milestone event” construction, which allows us to identify random times beyond which a random walk with negative drift will never go above a previously achieved level.

The “milestone events” are similar to the ladder height decomposition of a random walk, but we cannot directly use ladder height theory because the corresponding expressions for the probabilities of interest (for example the probability of an infinite strictly increasing ladder epoch) are rarely computable in closed form. The “milestone construction” introduces a parameter m which, together with change of measure ideas, allows to simulate without bias the occurrence of object such as the time the random walk reaches a certain barrier, for example.

Putting these “milestone events” of the random walks together and using the fact that $M(t)$ can be bounded by the appropriate running time maximums of the random walks, we can find the desired Δ_k . We next provide the details of the construction.

Decomposition Choose $a \in (\lambda, c\mu)$. Then, for $t > 0$,

$$X(t) = N^0(t) - \sum_{i=1}^c N^i(t) = (N^0(t) - at) + \sum_{i=1}^c \left(\frac{a}{c}t - N^i(t) \right).$$

We define $(c + 1)$ random walks with negative drift associated with $N^i(t)$ as follows:

$$S_0^{(0)} = 0, \quad S_1^{(0)} = -aT_1^0 + 1, \quad S_n^{(0)} = S_{n-1}^{(0)} + (-aA_{n-1} + 1) \text{ for } n \geq 2. \quad (3.16)$$

If particular, $S_n^{(0)} = N^0(T_n^0) - aT_n^0$. For $i = 1, \dots, c$,

$$S_0^{(i)} = \frac{a}{c}T_1^i, \quad S_n^{(i)} = S_{n-1}^{(i)} + \left(-1 + \frac{a}{c}V_n^i \right) \text{ for } n \geq 1. \quad (3.17)$$

Here, $S_n^{(i)} = N^i(T_n^i -) - aT_n^i$. Figure 3.2 plots the relationship between $\{N^0(t) - at : t \geq 0\}$ and $\{S_n^{(0)} : n \geq 0\}$, and the relationship between $\{\frac{a}{c}t - N^i(t) : t \geq 0\}$ and $\{S_n^{(i)} : n \geq 0\}$ for $i = 1, \dots, c$.

In particular, we notice from Figure 3.2 that

$$\max_{s \geq t} \{N^0(s) - as\} = \max \left\{ N^0(t) - at, \max_{n \geq N^0(t)+1} \{S_n^{(0)}\} \right\} \leq \max_{n \geq N^0(t)} \{S_n^{(0)}\},$$

and for $i = 1, \dots, c$,

$$\max_{s \geq t} \left\{ \frac{a}{c}s - N^i(s) \right\} = \max_{n \geq N^i(t_-)} \{S_n^{(i)}\} \leq \max_{n \geq (N^i(t)-1)_+} \{S_n^{(i)}\}.$$

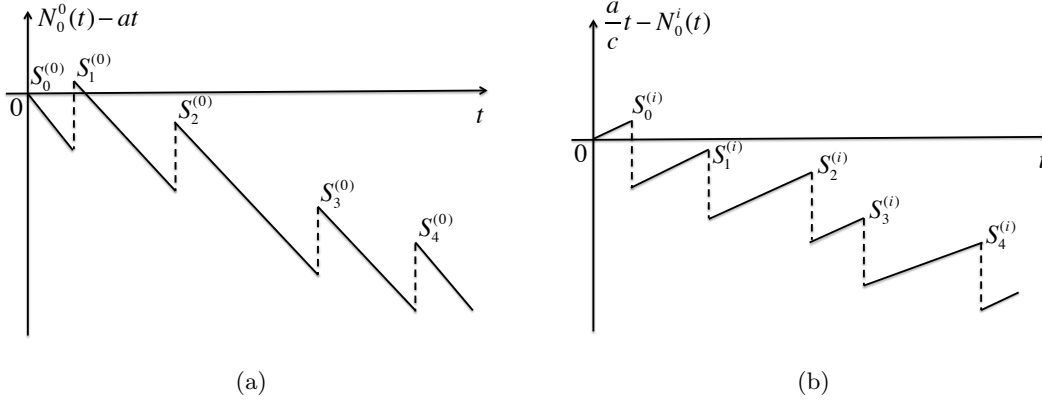


Figure 3.2: The relationship between the renewal processes and the random walks **(a)** $N^0(t) - at$ and $S_n^{(0)}$ **(b)** $(a/c)t - N^i(t)$ and $S_n^{(i)}$

We then notice that, for any given T ,

$$\begin{aligned} M(T) = \max_{t \geq T} \{X(t)\} &= \max_{t \geq T} \left\{ (N^0(t) - at) + \sum_{i=1}^c \left(\frac{a}{c}t - N^i(t) \right) \right\} \\ &\leq \max_{t \geq T} \{N^0(t) - at\} + \sum_{i=1}^c \max_{t \geq T} \left\{ \frac{a}{c}t - N^i(t) \right\} \\ &\leq \max_{n \geq N^0(T)} \{S_n^{(0)}\} + \sum_{i=1}^c \max_{n \geq N^i(T)-1} \{S_n^{(i)}\}. \end{aligned} \quad (3.18)$$

Milestone construction We use the “milestone events” construction to generate the $(c+1)$ random walks with negative drift, $S^{(i)}$, together with their running time maxima, $M_k^{(i)} := \max_{n \geq k} \{S_n^{(i)}\}$, $k \geq 0$, $i = 0, 1, \dots, c$. This construction is introduced in [Blanchet and Sigman, 2011; Blanchet and Wallwater, 2015], and we shall provide a brief overview here.

Fix $m > 0$ and $L \geq 1$ such that $P(m < M_0^{(i)} \leq (L+1)m) > 0$ for $i = 0, \dots, c$. The values of m and L do not seem to have significant impact on algorithm performance, as long as they are chosen to be small. In our numerical implementations, we choose $m = 1$ and $L = 3$.

For each random walk $\{S_n^{(i)} : n \geq 0\}$, $i = 0, 1, \dots, c$, we shall define a sequence of downward and upward “milestone events”, which we denoted as Φ_j^i and Υ_j^i , respectively, for $j \geq 0$ as follows:

$$\Phi_0^i := 0, \quad \Upsilon_0^i := 0,$$

and for $j \geq 1$,

$$\begin{aligned} \Phi_j^i &:= \inf \left\{ n \geq \Upsilon_{j-1}^i I(\Upsilon_{j-1}^i < \infty) \vee \Phi_{j-1}^i : S_n^{(i)} < S_{\Phi_{j-1}^i}^{(i)} - Lm \right\}, \\ \Upsilon_j^i &:= \inf \left\{ n \geq \Phi_j^i : S_n^{(i)} > S_{\Phi_j^i}^{(i)} + m \right\}. \end{aligned}$$

Notice that $P(\Phi_j^i < \infty) = 1$ while $P(\Upsilon_j^i < \infty) < 1$, as the random walks have negative drift. In fact, under Assumption 1, Proposition 2.1 in [Blanchet and Wallwater, 2015] shows $P(\Upsilon_j^i = \infty, i.o.) = 1$. We observe that when the event $\{\Upsilon_j^i = \infty\}$ happens, we know that the random walk will never go above $S_{\Phi_j^i}^{(i)} + m$ beyond Φ_j^i . This important observation allows us to find the running time maximum $M_k^{(i)}$. In particular, let $\Phi_{k^*}^i$ denote the first downward milestone at or after step k , and let $\Phi_{k^{**}}^i$ be the first downward milestone after $\Phi_{k^*}^i$ with $\Upsilon_{k^{**}}^i = \infty$. Then, after step $\Phi_{k^{**}}^i$, the random walk $S^{(i)}$ will never go above the level $S_{\Phi_{k^{**}}^i}^{(i)} + m$, and $S_{\Phi_{k^{**}}^i}^{(i)} + m < S_{\Phi_{k^*}^i}^{(i)} - Lm + m \leq S_{\Phi_{k^*}^i}^{(i)}$. Therefore, $M_k^{(i)} = \max_{n \geq k} \{S_n^{(i)}\} = \max_{k \leq n \leq \Phi_{k^{**}}^i} \{S_n^{(i)}\}$, i.e., we just need to find the maximum value of the random walk between step k and step $\Phi_{k^{**}}^i$. Figure 3.3 provides a pictorial explanation of the construction.

We are now ready to use the milestone events across the $(c+1)$ random walks to identify Δ_k associated with each T_k^0 ($k \geq 1$), such that $N^i(T_k^0) \geq 1$ for $i = 1, \dots, c$. Define

$$\Lambda_k^0 := \min_{j \geq 1} \left\{ \Phi_j^0 > N^0(T_k^0) : S_{\Phi_j^0}^{(0)} \leq S_{N^0(T_k^0)}^{(0)} - m, \Upsilon_j^0 = \infty \right\}, \quad (3.19)$$

and, for $i = 1, \dots, c$,

$$\Lambda_k^i := \min_{j \geq 1} \left\{ \Phi_j^i > N^i(T_k^0) - 1 : S_{\Phi_j^i}^{(i)} \leq S_{N^i(T_k^0)-1}^{(i)} - m, \Upsilon_j^i = \infty \right\}. \quad (3.20)$$

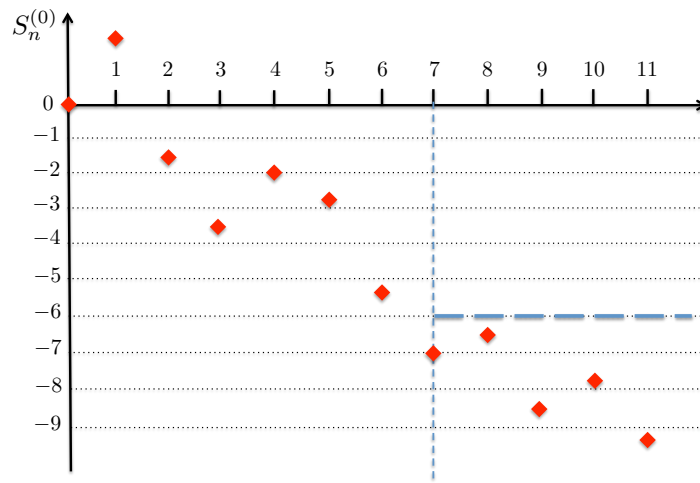


Figure 3.3: This figure plots a realization of the sample path $\{S_n^{(0)} : 0 \leq n \leq 11\}$. Here we set $m = 1$ and $L = 3$. Then, $\Phi_1^0 = 3$, $\Upsilon_1^0 = 4$, $\Phi_2^0 = 7$. If $\Upsilon_2^0 = \infty$, then for $n \geq 7$, $S_n^{(0)}$ will stay below the level $S_7^{(0)} + m$, which is demonstrated by the bold dashed line. Thus, $M_2^{(0)} = \max_{n \geq 2} \{S_n^{(0)}\} = S_2^{(0)}$ by only comparing the random walk values between step 2 and step 7.

In particular, the random walk $\{S_n^{(i)} : n \geq 0\}$ will never go above the level $S_{\Lambda_k^i}^{(i)} + m$ for $n \geq \Lambda_k^i$, $i = 0, \dots, c$. Let

$$\Delta_k := \max \left\{ T_{\Lambda_k^0}^0, \max_{1 \leq i \leq c} \left\{ T_{\Lambda_k^i}^i \right\} \right\} - T_k^0, \quad (3.21)$$

Since $N^0(T_k^0 + \Delta_k) \geq \Lambda_k^0$ and $N^i(T_k^0 + \Delta_k) - 1 \geq \Lambda_k^i$ for $i = 1, \dots, c$,

$$\max_{n \geq N^0(T_k^0 + \Delta_k)} \left\{ S_n^{(0)} \right\} \leq S_{\Lambda_k^0}^{(0)} + m \quad \text{and} \quad \max_{n \geq N^i(T_k^0 + \Delta_k) - 1} \left\{ S_n^{(i)} \right\} \leq S_{\Lambda_k^i}^{(i)} + m.$$

Therefore,

$$\begin{aligned} \max_{t \geq T_k^0 + \Delta_k} \{X(t)\} &\leq \max_{n \geq N^0(T_k^0 + \Delta_k)} \left\{ S_n^{(0)} \right\} + \sum_{i=1}^c \max_{n \geq N^i(T_k^0 + \Delta_k) - 1} \left\{ S_n^{(i)} \right\} \\ &\leq S_{\Lambda_k^0}^{(0)} + m + \sum_{i=1}^c \left(S_{\Lambda_k^i}^{(i)} + m \right) \\ &\leq S_{N^0(T_k^0)}^{(0)} + \sum_{i=1}^c S_{N^i(T_k^0) - 1}^{(i)} \\ &\leq N^0(T_k^0) - aT_k^0 + \sum_{i=1}^c \left(\frac{a}{c} T_k^0 - N^i(T_k^0) \right) \\ &= X(T_k^0) \leq \max_{T_k^0 \leq t \leq T_k^0 + \Delta_k} \{X(t)\}. \end{aligned}$$

Under Assumption 1, the time it takes to find Δ_k using the “milestone” construction has finite expectation (Theorem 2.2 in [Blanchet and Wallwater, 2015]). We shall provide the algorithmic details to generate the random walk with negative drift together with the “milestone” events for the light-tailed case in Section 3.3.1 to demonstrate the basic idea. The general case can be found in [Blanchet and Wallwater, 2015]. We also provide the algorithm to match the service time requirements to the customers in vacation system between two consecutive inspection times in Section 3.3.2. Lastly, the exact simulation algorithm of $GI/GI/c$ queue is summarized in Section 3.3.3.

3.3.1 Simulate a random walk with negative drift jointly with “milestone” events

To demonstrate the basic idea, we work with a generic random walk with negative drift $S_n := S_{n-1} + X_n$, for $n \geq 0$, with S_0 given. We also impose the light-tail assumption on X_n , i.e., there exist $\theta > 0$ such that $E[\exp(\theta X_n)] < \infty$. Let

$$\Phi_0 := 0, \quad \Upsilon_0 := 0,$$

and, for $j \geq 1$,

$$\begin{aligned} \Phi_j &:= \inf \{n \geq \Upsilon_{j-1} I(\Upsilon_{j-1} < \infty) \vee \Phi_{j-1} : S_n < S_{\Phi_{j-1}} - Lm\} \\ \Upsilon_j &:= \inf \{n \geq \Phi_j : S_n > S_{\Phi_j} + m\}. \end{aligned}$$

We also denote $\tau_m = \inf\{n \geq 0 : S_n > m, S_0 = 0\}$. Notice that $P(\Upsilon_j = \infty) = P(\tau_m = \infty) > 0$.

Sampling Φ_j is straightforward. We just sample the random walk, S_n , until $S_n < S_{\Phi_{j-1}} - Lm$. Sampling Υ_j and the path conditional on $\Upsilon_j < \infty$ requires more advanced simulation techniques, as $P(\Upsilon_j = \infty) > 0$. In particular, we use the exponential tilting idea discussed in [Asmussen, 2003]. Let $\psi_X(\theta) = \log E[\exp(\theta X_n)]$ be the log moment generating function of X_n , then we have $E[X_n] = \psi'_X(0) < 0$ and $\text{Var}(X_n) = \psi''_X(0) > 0$. By the convexity of $\psi_X(\cdot)$, we can always find $\eta > 0$ with $\psi_X(\eta) = 0$ and $\psi'_X(\eta) \in (0, \infty)$. Hence, we can define a new measure P_η based on exponential tilting so that

$$\frac{dP_\eta}{dP}(X_n) = \exp(\eta X_n).$$

Under P_η , S_n is a random walk with positive drift $\psi'_X(\eta)$. Thus $P_\eta(\tau_m < \infty) = 1$. By our choice of η , we also have $P(\tau_m < \infty) = E_\eta \exp(-\eta S_{\tau_m})$. In implementation, we shall generate the path S_n under P_η until τ_m and check whether $U \leq \exp(-\eta S_{\tau_m})$, where U is a uniform random variable independent of everything. If $U \leq \exp(-\eta S_{\tau_m})$, we claim that $\tau_m < \infty$ and accept the path $(S_n : n \leq \tau_m)$ as the path of the random walk conditional on $\tau_m < \infty$.

The algorithm to sample the random walk together with the milestone events goes as follows. *Throughout this thesis, “sample” in the pseudocode means sampling independently from everything that has already been sampled.*

Algorithm RWS: Sample a random walk with negative drift until stopping criteria are met

Input: $L, m, \{S_0, \dots, S_n\}, \{\Phi_0, \dots, \Phi_j\}, \{\Upsilon_0, \dots, \Upsilon_j\}$ and stopping criteria \mathcal{H} .

(Note that $n = \Phi_j$ if $\Upsilon_j = \infty$, and $n = \Upsilon_j$ otherwise. If there is no previous simulated partial random walk, then we initialize $n = 0, j = 0, \Phi_0 = 0, \Upsilon_0 = 0$, and S_0 as needed.)

1. While the stopping criteria \mathcal{H} are not satisfied, set $j \leftarrow j + 1$.

(a) (Downward milestone simulation)

Sample $\{S_k : n + 1 \leq k \leq \Phi_j\}$ under the nominal measure, i.e., generate the random walk until $S_n < S_{\Phi_{j-1}} - Lm$. Update $n = \Phi_j$.

(b) (Upward milestone simulation)

Sample $\tilde{S}_1, \dots, \tilde{S}_{\tau_m}$ from the tilted measure $P_\eta(\cdot)$. Sample $U \sim \text{Uniform}[0,1]$. If $U \leq \exp(-\eta\tilde{S}_{\tau_m})$, set $\Upsilon_j = n + \tau_m$, $S_{n+k} = S_n + \tilde{S}_k$ for $k = 1, \dots, \tau_m$ and update $n \leftarrow n + \tau_m$; otherwise set $\Upsilon_j = \infty$.

2. Output updated $\{S_0, \dots, S_n\}$, $\{\Phi_0, \dots, \Phi_j\}$ and $\{\Upsilon_0, \dots, \Upsilon_j\}$.

3.3.2 Simulate the vacation system between inspection times

To summarize our discussion above, in this section, we provide the pseudocode for generating the vacation system between the inspection time $T_{\kappa_l}^0$ and $T_{\kappa_{l+1}}^0$, for $l \geq 0$, $\kappa_0 = 0$, and $\kappa_{l+1} < \kappa_l$.

Algorithm VSS: Sample vacation system between $T_{\kappa_l}^0$ and $T_{\kappa_{l-1}}^0$, and extract corresponding service times

Input: $m, L, \kappa_l, \kappa_{l-1}, \{S_0^{(i)}, \dots, S_{n_i}^{(i)}\}, \{\Phi_0^i, \dots, \Phi_{j_i}^i\}, \{\Upsilon_0^i, \dots, \Upsilon_{j_i}^i\}$ for $i = 0, 1, \dots, c$.

1. Apply Algorithm RWS to further sample $S^{(0)}$ with the stopping criteria \mathcal{H} being $n_0 \geq |\kappa_l|$. Then, find $T_{|\kappa_l|}^0$.
2. Apply Algorithm RWS to further sample $S^{(0)}$ with the stopping criteria \mathcal{H} being $n_0 = \Lambda_{|\kappa_l|}^0$, with $\Lambda_{|\kappa_l|}^0$ defined in Eq. (3.19).
3. For $i = 1, \dots, c$, apply Algorithm RWS to further sample $S^{(i)}$ until the stopping criteria \mathcal{H} being $n_i = \Lambda_{|\kappa_l|}^i$, with $\Lambda_{|\kappa_l|}^i$ defined in Eq. (3.20).
4. Compute $\Delta_{|\kappa_l|}$ as defined in Eq. (3.21). For $i = 0, 1, \dots, c$, apply Algorithm RWS to further sample $S^{(i)}$ with the stopping criteria \mathcal{H} being $T_{n_i}^i \geq T_{|\kappa_l|}^0 + \Delta_{|\kappa_l|}$.
5. Construct the backward renewal processes $\{N^i(t) : T_{\kappa_l}^0 - \Delta_{|\kappa_l|} \leq t \leq 0\}$ using $\{S_n^{(i)} : 0 \leq n \leq n_i\}$ for $i = 0, 1, \dots, c$. In particular, we shall set $T_{-n}^i = -T_n^i$. Then, construct $X(t) = N^0(t) - \sum_{i=1}^c N^i(t)$ for $t \in [T_{\kappa_l}^0 - \Delta_{|\kappa_l|}, 0]$.

6. Set $M(T_{\kappa_l}^0) = \max_{T_{\kappa_l}^0 - \Delta_{|\kappa_l|} \leq t \leq T_{\kappa_l}^0} \{X(t)\}$ and then compute $Q_v(T_{\kappa_l}^0) = M(T_{\kappa_l}^0) - X(T_{\kappa_l}^0)$ to be the number of people waiting in the queue at time $T_{\kappa_l}^0$. The remaining activity times are $U^i(T_{\kappa_l}^0)$, for $i = 1, \dots, c$.
7. If $Q_v(T_{\kappa_l}^0) > 1$, then for $1 \leq j \leq Q_v(T_{\kappa_l}^0) - 1$, the j -th people waiting in queue arrive at time $T_{\kappa_l - Q_v(T_{\kappa_l}^0) + j}^0$. Let $\tilde{D}_j = \inf\{t \geq 0 : j = \sum_{i=1}^c \sigma_{T_{\kappa_l}^0}^i(t)\}$, then extract $V_{\kappa_l - Q_v(T_{\kappa_l}^0) + j} = \sum_{i=1}^c V_{N^i(T_{\kappa_l}^0 + \tilde{D}_j)} dN^i(T_{\kappa_l}^0 + \tilde{D}_j)$ as his service time.
8. For $\kappa_l \leq n \leq -1$, use Eq. (3.3) to extract their service times $V_{\kappa_l}, \dots, V_{-1}$.
9. Output
 - (a) service times of the people waiting in queue at time $T_{\kappa_l}^0$ (excluding the arrival at $T_{\kappa_l}^0$), i.e., null if $Q_v(T_{\kappa_l}^0) = 1$ and $(V_{\kappa_l - Q_v(T_{\kappa_l}^0) + 1}, \dots, V_{\kappa_l - 1})$ in the order of arrivals if $Q_v(T_{\kappa_l}^0) > 1$.
 - (b) matched arrival times and service times $\left\{ (T_j^0, V_j) : \kappa_l \leq j \leq -1 \right\}$ in the order of arrival.
 - (c) updated random walks $\{S_0^{(i)}, \dots, S_{n_i}^{(i)}\}$ with updated milestone events $\{\Phi_0^i, \dots, \Phi_{j_i}^i\}$, $\{\Upsilon_0^i, \dots, \Upsilon_{j_i}^i\}$ for $i = 0, 1, \dots, c$.

3.3.3 Overall exact simulation procedure

In this section, we provide the overall pseudocode for our exact simulation algorithm.

Algorithm PS: sample stationary $GI/GI/c$ queue at time 0

Input: m, L, F, G, c

1. For $i = 0, 1, \dots, c$, initiate $\Phi_0^i = \Upsilon_0^i = 0$, and $S_0^{(i)}$ as defined in Eqs. (3.16, 3.17).
2. Set $\kappa_0 = 0, \kappa_1 = -10, l = 1$.
3. (a) Apply Algorithm VSS to sample vacation system between $T_{\kappa_l}^0$ and $T_{\kappa_l - 1}^0$, and extract corresponding service times.
 - (b) Start two $GI/GI/c$ queues, both from $T_{\kappa_l}^0$, one initialized with

$$(Q_v(T_{\kappa_l}^0), \mathcal{S}(U(T_{\kappa_l}^0)), 0)$$

and the other initialized with $\mathbf{0}$. Evolve the two queues forward in time until time 0 and calculate

$$C = \min_{\kappa_l \leq j \leq -1} \|Z_{T_{\kappa_l}^0}(T_j^0 - T_{\kappa_l}^0; (Q_v(T_{\kappa_l}^0), \mathcal{S}(U(T_{\kappa_l}^0))), 0)) - Z_{T_{\kappa_l}^0}(T_j^0 - T_{\kappa_l}^0; \mathbf{0})\|_{\infty}.$$

4. If $C = 0$, output $Z(0) = Z_{T_{\kappa_l}^0}(|T_{\kappa_l}^0|; \mathbf{0})$. Otherwise ($C > 0$), set $l \leftarrow l + 1$, $\kappa_l = 2\kappa_{l-1}$, then go back to Step 3.

3.4 Numerical experiments

As a sanity check, we have implemented our MATLAB code in the case of an Erlang(k_1, λ)/Erlang(k_2, μ)/ c queue.

Firstly, in the context of the $M/M/c$ queue, which is a special case of Erlang(k_1, λ)/Erlang(k_2, μ)/ c when $k_1 = k_2 = 1$ and whose stationary distribution can be computed in closed form, we have compared the theoretical distribution to the empirical distribution of the number of customers in the system at stationarity. The empirical distribution is produced from a large number of runs using our perfect simulation algorithm. Figure 3.4 shows a comparison of these distributions when $\lambda = 3$, $\mu = 2$ and $c = 2$. Grey bars show the empirical result of 5000 draws using our perfect simulation algorithm, and black bars show the theoretical distribution of the number of customers in the system. The two are very close to each other. Following [Connor and Kendall, 2015], we test the goodness of fit using a Pearson's chi-squared test; under the null hypothesis, the empirical histogram converges to theoretical distribution as the sample size increases. The test yields a p -value equal to 0.6806, indicating close agreement (i.e., we can not reject the null hypothesis). Similarly, Figure 3.5 provides another comparison with a different set of parameters, $\lambda = 10$, $\mu = 2$, $c = 10$, with a p -value being 0.6454 from the chi-squared test.

Also, for a general Erlang(k_1, λ)/Erlang(k_2, μ)/ c queue ($k_1 > 1, k_2 > 1$) when traffic intensity $\rho = (\lambda_1 k_2) / (c \lambda_2 k_1) = 0.9$, we have compared the empirical distribution obtained from simulation with the numerical results (with precision at least 10^{-4}) provided in Table III of [Hillier and Lo, 1971]. Figure 3.6 shows the comparison for an $E_3/E_2/5$ queue with $\rho = 0.9$. We observe that the

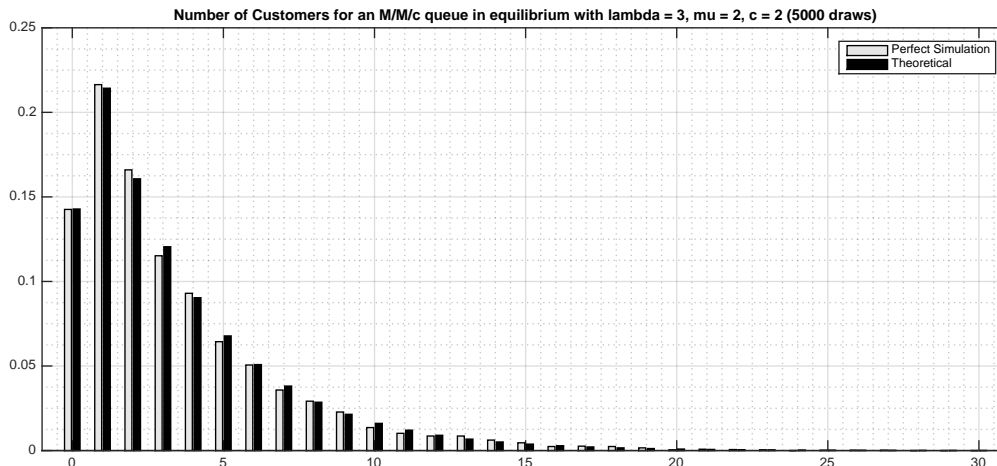


Figure 3.4: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 3$, $\mu = 2$ and $c = 2$.

two histograms are very close to each other. A Pearson's chi-squared test between the simulated distribution and the numerical one gives a p -value of 0.6815.

Next, we run numerical experiments in $M/M/c$ case to see how the running time of our algorithm, measured by mean coalescence time of two bounding systems, scales as the number of servers grows and the traffic intensity ρ changes. Starting from time 0, the upper bound queue has its queue length sampled from the theoretical distribution of an $M/M/c$ vacation system and all servers busy with remaining service times drawn from the equilibrium distribution of the service/vacation time; and the lower bound queue is empty. Then, we run both the upper bound and lower bound queues forward in time with the same stream of arrival times and service requirements until they coalesce. Table 3.1 shows the estimated mean coalescence time, $E[T]$, based on 5000 iid samples, for different system scales in the quality-driven regime (QD). We observe that $E[T]$ does not increase much as the system scale parameter, s , grows. Table 3.2 shows similar results for the quality-and-efficiency driven operating regime (QED). In this case, $E[T]$ increases at a faster rate with s than the QD case, but the magnitude of increment is still not significant.

Finally we run a numerical experiment in the $M/M/c$ case aiming to test how computational complexity of our algorithm changes with traffic intensity, $\rho = \lambda/(c\mu)$. Here, we define the computational complexity as the total number of renewals (including arrivals and services/vacations)

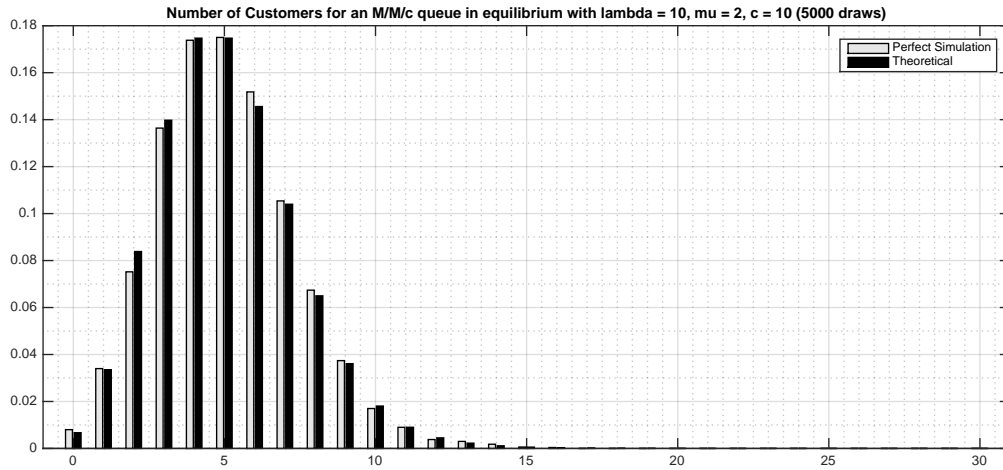


Figure 3.5: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 10$, $\mu = 2$ and $c = 10$.

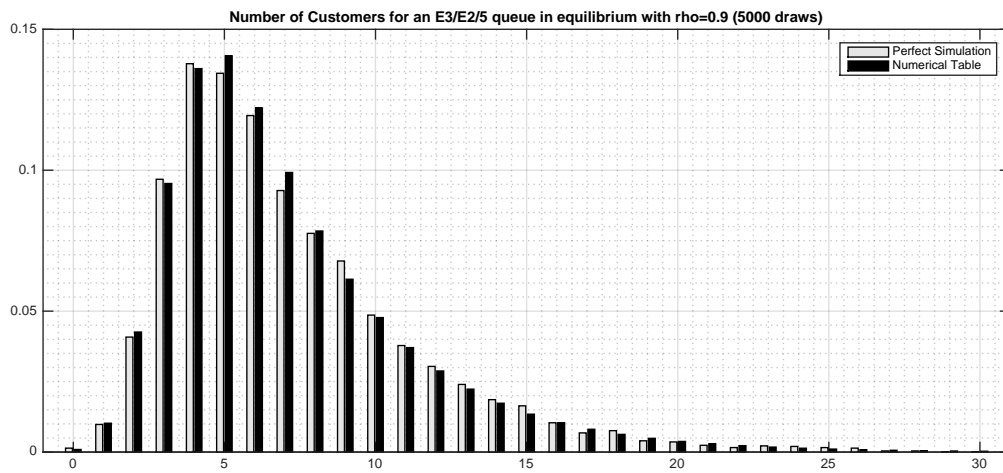


Figure 3.6: Number of customers for an $\text{Erlang}(k_1, \lambda)/\text{Erlang}(k_2, \mu)/c$ queue in stationarity when $k_1 = 3$, $\lambda = 4.5$, $k_2 = 2$, $\mu = 2/3$, $c = 5$ and $\rho = 0.9$.

Table 3.1: Simulation estimates for the mean coalescence time of $M/M/c$ queue (QD)

$$(\lambda_s = s, c_s = 1.2s, \mu = 1)$$

s	Mean	95% confidence interval
100	6.4212	[6.2902, 6.5522]
500	7.0641	[6.9848, 7.1434]
1000	7.7465	[7.6667, 7.8263]

Table 3.2: Simulation estimates for the mean coalescence time of $M/M/c$ queue (QED)

$$(\lambda_s = s, c_s = s + 2\sqrt{s}, \mu = 1)$$

s	Mean	95% confidence interval
100	6.5074	[6.3771, 6.6377]
500	8.5896	[8.4361, 8.7431]
1000	9.4723	[9.3041, 9.6405]

the algorithm samples in total to find the coalescence time. We expect the complexity to scale like $(c + 1)(1 - \rho)^{-2}E[T(\rho)]$, where $(c + 1)$ is the number of renewal processes we need to simulate, $(1 - \rho)^{-2}$ is on average the amount of renewals we need to sample to find its running time maximum for each renewal process, and $E[T(\rho)]$ is the mean coalescence time when the traffic intensity is ρ . Table 3.3 summarizes our numeral results, based 5000 independent runs of the algorithm for each ρ . We run the coalescence check at $\kappa = 10 \times 2^k$, for $k = 1, 2, \dots$, until we find the coalescence. We observe that as ρ increase, the computational complexity increases significantly, but when multiplied by $(1 - \rho)^2$, the resulting products are of about the same magnitude – up to a factor proportional to λ , given that the number of arrivals scales as λ per unit time. Therefore, the main scaling parameter for the complexity here is $(1 - \rho)^{-2}$. Notice that if we simulate the system forward in time from empty, it also took around $O((1 - \rho)^{-2})$ arrivals to get close to stationary.

Table 3.3: Simulation result for computational complexities with varying traffic intensities

 $M/M/c$ queue with fixed $\mu = 5$ and $c = 2$

λ	traffic intensity (ρ)	mean number of renewals sampled	mean index of successful inspection time	mean number of renewals sampled $\times (1 - \rho)^2$
5	0.5	225.6670	11.7780	56.4168
6	0.6	377.0050	14.7780	60.3208
7	0.7	764.3714	21.9800	68.7934
8	0.8	2,181.3452	44.2320	87.2538
9	0.9	12,162.6158	161.0840	121.6262

Chapter 4

Exact Simulation with Random Assignment

We give our second set of exact simulation algorithms, which utilizes dominated coupling from the past (DCFTP) protocol and the random assignment (RA) discipline, in this chapter. In Section 4.1, we give some theoretical background of multi-server queues under FIFO and RA, and establish the dominance relationship of workload between these two different service disciplines. In Section 4.2, we describe our simulation strategies and the main theoretic result. In Section 4.3, we provide the numerical experiments results for sanity check and performance comparison. In Sections 4.4 and 4.5, we extend the developed algorithm to perform perfect sampling for various other queueing settings. Detailed simulation algorithm steps and the proof of the main technical result are provided in Appendix B.

4.1 The FIFO and RA $GI/GI/c$ model

4.1.1 The FIFO $GI/GI/c$ model

In what follows, as input to a c -server in parallel multi-server queue, we have iid service times $\{V_n : n \geq 0\}$ distributed as $F(x) = P(V \leq x)$, $x \geq 0$, with finite and non-zero mean $0 < E[V] = 1/\mu < \infty$. Independently, the arrival times $\{t_n : n \geq 0\}$ ($t_0 = 0$) to the model form a renewal process with iid interarrival times $A_n = t_{n+1} - t_n$, $n \geq 0$ distributed as $G(x) = P(A \leq x)$, $x \geq 0$,

and finite non-zero arrival rate $0 < \lambda = E[A]^{-1} < \infty$. The FIFO $GI/GI/c$ model has only one queue (line), and we let $W_n = (W_n(1), \dots, W_n(c))^T$ denote the *Kiefer-Wolfowitz workload vector* (see, for example, Page 341 in Chapter 12 of [Asmussen, 2003]). It satisfies the recursion

$$W_{n+1} = \mathcal{S}(W_n + V_n \mathbf{e}_1 - A_n \mathbf{1})^+, \quad n \geq 0, \quad (4.1)$$

where $\mathbf{e}_1 = (1, 0, \dots, 0)^T$, $\mathbf{1} = (1, 1, \dots, 1)^T$, \mathcal{S} places a vector in ascending order, and $^+$ takes the positive part of each coordinate, as we mentioned earlier in Eq. (3.4). Let C_n denote the n^{th} arriving customer. For $i = 1, \dots, c$, $W_n(i)$ is the waiting time of C_n if he decides to enter service immediately after there are at least i servers available once he reaches the head of the queue, i.e., $D_n = W_n(1)$ is the customer delay in queue (line) of C_n . Recursion (4.1) defines a Markov chain due to the given iid assumptions.

With stability condition $\rho = \lambda/(c\mu) < 1$, it is well known that W_n converges in distribution as $n \rightarrow \infty$ to a proper stationary distribution. Let π denote this stationary distribution. Again, our main objective is to provide a simulation algorithm for sampling exactly from π .

4.1.2 The RA $GI/GI/c$ model

Given a c -server queueing system, the random assignment model (RA) is the case when each of the c servers forms its own FIFO single-server queue, and each arrival to the system, independent of the past, randomly chooses queue i to join with equal probability $1/c$, $1 \leq i \leq c$. In the $GI/GI/c$ case, we refer to this as the RA $GI/GI/c$ model. The following is a special case of Lemma 1.3, Page 342 in [Asmussen, 2003]. Such results and others even more general are based on [Wolff, 1987], [Foss, 1980], and [Foss and Chernova, 2001].

Lemma 4. *Let $Q_F(t)$ denote total number of customers in system at time $t \geq 0$ for the FIFO $GI/GI/c$ model, and let $Q_{RA}(t)$ denote total number of customers in system at time $t \geq 0$ for the corresponding RA $GI/GI/c$ model in which both models are initially empty and fed with exactly the same input of renewal arrivals $\{t_n : n \geq 0\}$ and iid service times $\{V_n : n \geq 0\}$. Assume further that for both models the service times are used by the servers in the order in which service initiations occur (V_n is the service time used for the n -th such initiation). Then,*

$$P(Q_F(t) \leq Q_{RA}(t), \text{ for all } t \geq 0) = 1. \quad (4.2)$$

The importance of Lemma 4 is that it allows us to jointly simulate versions of the two stochastic processes $\{Q_F(t) : t \geq 0\}$ and $\{Q_{RA}(t) : t \geq 0\}$ while achieving a coupling such that (4.2) holds. In particular, whenever an arrival finds the RA model empty, the FIFO model is found empty as well. (But we need to impose further conditions if we wish to ensure that indeed the RA $GI/GI/c$ queue will empty with certainty.) Letting time t be sampled at arrival times of customers, $\{t_n : n \geq 0\}$, we thus also have

$$P(Q_F(t_n-) \leq Q_{RA}(t_n-), \text{ for all } n \geq 0) = 1. \quad (4.3)$$

In other words, the total number in system as found by the n -th arrival is sample-path ordered as well. Note that for the FIFO model, the n -th arriving customer C_n initiates the n -th service since FIFO means “First-In-Queue-First-Out-of-Queue” where by “queue” we mean the line before entering service. This means that for the FIFO model we can attach V_n to C_n upon arrival if we so wish when applying Lemma 4. For the RA model, however, customers are not served in the order they arrive. For example, consider $c = 2$ servers (system initially empty) and suppose C_1 is assigned to server 1 with service time V_1 , and C_2 also is assigned to server 1 (before C_1 departs) with service time V_2 . Meanwhile, before C_1 departs, suppose C_3 arrives and is assigned to the empty server 2 with service time V_3 . Then, V_3 is used for the second service initiation. *For RA, the service times in the order of initiation are a random permutation of the originally assigned $\{V_n\}$.*

To use Lemma 4, it is crucial to simply let the servers hand out service times one at a time when they are needed for a service initiation. Thus, customers waiting in a queue before starting service do not have a service time assigned until they enter service. In simulation terminology, this amounts to generating the service times in order of when they are needed.

One disadvantage of generating service times only when they are needed, is that it then does not allow workload¹ to be defined; only the amount of work in service. To get around this if need be, one can simply generate service times upon arrival of customers, and give them to the servers to be used in the order of service initiation. The point is that when C_n arrives, the total work in system jumps up by the amount V_n . But V_n is not assigned to C_n , it is assigned (perhaps later) to which ever customer initiates the n -th service. This allows Lemma 4 to hold true for total amount

¹Workload (total) at any time t is defined as the sum of all whole and remaining service times in the system at time t .

of work in the system: If we let $\{\nu_F(t) : t \geq 0\}$ and $\{\nu_{RA}(t) : t \geq 0\}$ denote total workload in the two models with the service times used in the manner just explained, then in addition to Lemma 4 we have

$$P(\nu_F(t) \leq \nu_{RA}(t), \text{ for all } t \geq 0) = 1, \quad (4.4)$$

$$P(\nu_F(t_n-) \leq \nu_{RA}(t_n-), \text{ for all } n \geq 0) = 1. \quad (4.5)$$

It is important, however, to note that what one can't do is define workload at the individual server i by doing this, because that forces us to assign V_n to C_n so that workload at the server that C_n attends (i say) jumps by V_n and C_n enters service using V_n ; that destroys the proper coupling needed to obtain Lemma 4. We can only handle the total (sum over all c nodes) workload. In the present chapter, our use of Lemma 4 is via a kind of reversal:

Lemma 5. *Let $\{V'_n\}$ be an iid sequence of service times distributed as F , and assign V'_n to C_n in the RA model. Define V_n as the service time used in the n -th service initiation. Then, $\{V_n\}$ is also iid distributed as F .*

Proof. The key is noting that we are re-ordering based only on the order in which service times begin being used, not when they are completed (which would thus introduce a bias). The service time chosen for the next initiation either enters service immediately (e.g., is one that is routed to an empty queue by an arriving customer) or is chosen from among those waiting in lines, and all those waiting are iid distributed as F . Let \hat{t}_n denote the time at which the n -th service initiation begins. The value V_n of the n -th service time chosen (at time \hat{t}_n) by a server is independent of the past service time values used before time \hat{t}_n , and is distributed as F (the choice of service time chosen as the next to be used is not based on the value of the service time, only its position in the lines). Letting $k(n) =$ the index of the $\{V'_n\}$ that is chosen, i.e., $V_n = V'_{k(n)}$, it is this index (a random variable) that depends on the past, but the value V_n is independent of $k(n)$ since it is a new one. Thus, $\{V_n\}$ are iid distributed as F . \square

The point of the above Lemma 5 is that we can, if we so wish, simulate the RA model by assigning V'_n to C_n (to be used as their service time), but then assigning V_n , i.e. $V'_{k(n)}$, to C_n in the FIFO model. By doing so the requirements of Lemma 4 are satisfied and Eqs. (4.2, 4.3, 4.4, 4.5) hold. Interestingly, however, it is not possible to first simulate the RA model up to a fixed time t ,

and then stop and reconstruct the FIFO model up to this time t : At time t , there may still be RA customers waiting in lines and hence not enough of the V_n have been determined yet to construct the FIFO model. But all we have to do, if need be, is to continue the simulation of the RA model beyond time t until enough V_n have been determined to construct fully the FIFO model up to time t .

4.2 Simulating exactly from the stationary distribution of the RA $GI/GI/c$ model

By Lemma 4, the RA $GI/GI/c$ queue, which shares the same arrival stream $\{t_n : n \geq 0\}$ ($t_0 = 0$) and same service times in the order of service initiations $\{V_n : n \geq 0\}$, will serve as a sample path upper bound (in terms of total number of customers in system and total workload) of the target FIFO $GI/GI/c$ queue. Independent of $\{A_n : n \geq 0\}$ and $\{V_n : n \geq 0\}$, we let $\{U_n : n \geq 0\}$ be an iid sequence of random variables from discrete uniform distribution on $\{1, 2, \dots, c\}$; U_n represents the choice that customer C_n makes about which single-server queue to join under RA discipline. Let $\bar{W}_n = (\bar{W}_n(1), \dots, \bar{W}_n(c))^T$ denote the workload vector as found by C_n in the RA $GI/GI/c$ model, and for $i = 1, \dots, c$, $\bar{W}_n(i)$ is the waiting time of C_n if he chooses to join the FIFO single-server queue of server i . So, $\bar{W}_0(i) = 0$ and

$$\bar{W}_{n+1}(i) = (\bar{W}_n(i) + V_n I(U_n = i) - A_n)^+, \quad n \geq 0. \quad (4.6)$$

These c processes are dependent through the common arrival times $\{t_n : n \geq 0\}$ (equivalently common interarrival times $\{A_n : n \geq 0\}$) and the common $\{U_n : n \geq 0\}$ random variables. Because of all the iid assumptions, $\{\bar{W}_n : n \geq 0\}$ forms a Markov chain. Define $\tilde{V}_n = (\tilde{V}_n(1), \dots, \tilde{V}_n(c))^T = (V_n I(U_n = 1), \dots, V_n I(U_n = c))^T$, then we can express (4.6) in vector form as

$$\bar{W}_{n+1} = (\bar{W}_n + \tilde{V}_n - A_n \mathbf{1})^+, \quad n \geq 0. \quad (4.7)$$

\bar{W}_n uses the same interarrival times $\{A_n : n \geq 0\}$ and service times $\{V_n : n \geq 0\}$ as we fed W_n in (4.1), however the coordinates of \bar{W}_n are not in ascending order, though all of them are nonnegative.

Each node i as expressed in (4.6) can be viewed as a FIFO $GI/GI/1$ queue with common renewal arrival process $\{t_n : n \geq 0\}$, but with iid service times $\{\tilde{V}_n(i) = V_n I(U_n = i) : n \geq 0\}$. Across

i , the service times $(\tilde{V}_n(1), \dots, \tilde{V}_n(c))$ are not independent, but they are identically distributed: marginally, with probability $1/c$, $\tilde{V}_n(i)$ is distributed as F , and with probability $(c-1)/c$ it is distributed as the point mass at 0; i.e., $E[\tilde{V}(i)] = E[V]/c$. The point here is that we are not treating node i as a single-server queue endowed only with its own arrivals (a thinning of the $\{t_n : n \geq 0\}$ sequence) and its own service times iid distributed as F . Defining iid increments $\Delta_n(i) = \tilde{V}_n(i) - A_n$ for $n \geq 0$, each node i has an associated random walk with negative drift $\{S_n(i) : n \geq 0\}$, where $S_0(i) = 0$ and

$$S_n(i) = \sum_{j=1}^n \Delta_j(i), \quad n \geq 1. \quad (4.8)$$

With $\rho = \lambda E[V]/c < 1$, we define $\rho_i = \lambda E[\tilde{V}(i)] = \lambda E[V]/c = \rho < 1$; equivalently $E[\Delta(i)] < 0$ for all $i = 1, \dots, c$. Let $\bar{W}^0(i)$ denote a random variable with the limiting (stationary) distribution of $\bar{W}_n(i)$ as $n \rightarrow \infty$, it is well known (due to the iid assumptions) that $\bar{W}^0(i)$ has the same distribution as

$$M(i) := \max_{m \geq 0} S_m(i)$$

for $i = 1, \dots, c$.

More generally, even when the increment sequence is just stationary ergodic, not necessarily iid (hence not time reversible as in the iid case), it is the backward in time maximum that is used in constructing a stationary version of $\{\bar{W}_n(i)\}$. We will need this backwards approach in our simulation so we go over it here; it is usually referred to as *Loynes' Lemma*. We extend the arrival point process $\{t_n : n \geq 0\}$ to be a two-sided point stationary renewal process $\{t_n : n \in \mathbb{Z}\}$

$$\dots t_{-2} < t_{-1} < 0 = t_0 < t_1 < t_2 \dots$$

Equivalently, $A_n = t_{n+1} - t_n$, $n \in \mathbb{Z}$, form iid interarrival times; $\{A_n : n \in \mathbb{Z}\}$ forms a two-sided iid sequence.

Similarly, the iid sequences $\{V_n : n \geq 0\}$ and $\{U_n : n \geq 0\}$ are extended to be two-sided iid, $\{V_n : n \in \mathbb{Z}\}$ and $\{U_n : n \in \mathbb{Z}\}$. These extensions further allow two-sided extension of the iid increment sequences $\{\Delta_n(i) : n \in \mathbb{Z}\}$ for $i = 1, \dots, c$, i.e.,

$$\Delta_n(i) = \tilde{V}_n - A_n = V_n I(U_n = i) - A_n, \quad n \in \mathbb{Z}.$$

Then, we define c time-reversed (increments) random walks $\{S_n^{(r)}(i) : n \geq 0\}$ for $i = 1, \dots, c$, by $S_0^{(r)}(i) = 0$ and

$$S_n^{(r)}(i) = \sum_{j=1}^n \Delta_{-j}(i), \quad n \geq 1. \quad (4.9)$$

A (from the infinite past) stationary version of $\{\bar{W}_n(i)\}$ denoted by $\{\bar{W}_n^0(i) : n \leq 0\}$ is then constructed via

$$\bar{W}_0^0(i) = \max_{m \geq 0} S_m^{(r)}(i), \quad (4.10)$$

$$\bar{W}_{-1}^0(i) = \max_{m \geq 1} S_m^{(r)}(i) - S_1^{(r)}(i), \quad (4.11)$$

$$\bar{W}_{-2}^0(i) = \max_{m \geq 2} S_m^{(r)}(i) - S_2^{(r)}(i), \quad (4.12)$$

⋮

$$\bar{W}_{-n}^0(i) = \max_{m \geq n} S_m^{(r)}(i) - S_n^{(r)}(i), \quad (4.13)$$

for all $i = 1, \dots, c$.

By construction, the process $\{\bar{W}_n^0 = (\bar{W}_n^0(1), \dots, \bar{W}_n^0(c))^T : n \leq 0\}$, is jointly stationary representing a (from the infinite past) stationary version of $\{\bar{W}_n : n \leq 0\}$, and satisfies the forward-time recursion (4.7):

$$\bar{W}_{n+1}^0 = \left(\bar{W}_n^0 + \tilde{V}_n - A_n \mathbf{1} \right)^+, \quad n \leq -1. \quad (4.14)$$

Thus, by starting at $n = 0$ and walking backward in time, we have (theoretically) a time-reversed copy of the RA model. Furthermore, $\{\bar{W}_n^0 : n \leq 0\}$ can be extended to include forward time $n \geq 1$ via using the recursion further:

$$\bar{W}_n^0 = \left(\bar{W}_{n-1}^0 + \tilde{V}_{n-1} - A_{n-1} \mathbf{1} \right)^+, \quad n \geq 1, \quad (4.15)$$

where $\tilde{V}_n = (V_n I(U_n = 1), \dots, V_n I(U_n = c))^T$ for $n \in \mathbb{Z}$.

In fact once we have a copy of just \bar{W}_0^0 , we can start off the Markov chain with it as initial condition and use (4.15) to obtain a forward in time stationary version $\{\bar{W}_n^0 : n \geq 0\}$.

The above “construction”, however, is theoretical. We do not yet have any explicit way of obtaining a copy of \bar{W}_0^0 , let alone an entire from-the-infinite-past sequence $\{\bar{W}_n^0 : n \leq 0\}$. In [Blanchet and Wallwater, 2015], a simulation algorithm is given that yields (when applied to each of our random walks), for each $1 \leq i \leq c$, a copy of $\{(S_n^{(r)}(i), \bar{W}_{-n}^0(i)) : 0 \leq n \leq N\}$ for any desired

$0 \leq N < \infty$ including N being stopping times. We modify the algorithm so that it can do the simulation jointly across the c systems, that is, we extend it to a multi-dimensional form.

In particular, it yields an algorithm for obtaining a copy of \bar{W}_0^0 , as well as a finite segment (of length N) of a backward in time copy of the RA model; $\{\bar{W}_{-n}^0 : 0 \leq n \leq N\}$, a stationary into the past construction up to discrete time $n = -N$.

Finite exponential moments are not required (because only *truncated* exponential moments are needed $E(e^{\gamma \Delta(i)} I\{|\Delta(i)| \leq a\})$, which in turn allow for the simulation of the exponential tilting of truncated $\Delta(i)$, via acceptance-rejection). To get finite expected termination time at each individual node, one needs the service distribution to have finite moment slightly beyond 2: For some (explicitly known) $\epsilon > 0$,

$$E[V^{2+\epsilon}] < \infty. \quad (4.16)$$

As our first case, we will be considering a stopping time N such that $\bar{W}_{-N} = \mathbf{0}$. Before we give the definition of the stopping time N , we introduce the main idea of our simulation algorithm.

Let us define the maximum of a sequence of vectors. Suppose we have k vectors Z_1, \dots, Z_k , where $Z_i \in \mathbb{R}^d$ with $d \geq 1$ and $k \in \mathbb{N}_+ \cup \{\infty\}$, define

$$\max(Z_1, \dots, Z_k) = \left(\max_{1 \leq i \leq k} Z_i(1), \dots, \max_{1 \leq i \leq k} Z_i(d) \right)^T.$$

Next define, for $n \in \mathbb{Z}$, that

$$\mathbf{u}_n = (I(U_n = 1), \dots, I(U_n = c))^T \quad \text{and} \quad \Delta_n = \tilde{V}_n - A_n \mathbf{1} = V_n \mathbf{u}_n - A_n \mathbf{1},$$

where $\{U_n : n \in \mathbb{Z}\}$ are iid from discrete uniform distribution over $\{1, 2, \dots, c\}$, and independently $\{A_n : n \in \mathbb{Z}\}$ are iid from distribution G (as introduced in Section 4.1.1). Our goal is to simulate the stopping time $N \in \mathbb{N}$ such that $\bar{W}_{-N}^0 = \mathbf{0}$, defined as

$$N = \inf\{n \geq 0 : \bar{W}_{-n}^0 = \max_{k \geq n} S_k^{(r)} - S_n^{(r)} = \mathbf{0}\}, \quad (4.17)$$

i.e., the first time walking in the past, that all coordinates of the workload vector are 0, jointly with $\{(S_n^{(r)}, \bar{W}_{-n}^0) : 0 \leq n \leq N\}$. (By convention, the value of any empty sum of numbers is zero, i.e., $\sum_{j=1}^0 a_j = 0$.)

To ensure that $E[N] < \infty$, in addition to $\rho < 1$ (stability), it is required that $P(A > V) > 0$ (see the proof of Theorem 2 in [Sigman, 1988]), for which the most common sufficient conditions

are that A has unbounded support, $P(A > t) > 0$, $t \geq 0$, or V has mass arbitrarily close to 0, $P(V < t) > 0$, $t > 0$. But as we shall show in Section 4.2.2, given we know that $P(A > V) > 0$, we can assume without loss of generality that interarrival times are bounded. It is that assumption which makes the extension of [Blanchet and Wallwater, 2015] to a multidimensional form easier to accomplish. Then, we show (in Section 4.2.3 and Section 4.6) how to still simulate from π even when $P(A > V) = 0$. We do that in two different ways, one as sandwiching argument and the other involving Harris recurrent Markov chain regenerations.

4.2.1 Algorithm for simulating exactly from π for the FIFO $GI/GI/c$ queue:

The case $P(A > V) > 0$

As mentioned earlier, we will assume that $P(A > V) > 0$, so that the stable ($\rho < 1$) RA and FIFO $GI/GI/c$ Markov chains (4.7) and (4.1) will visit $\mathbf{0}$ infinitely often with certainty. (That the RA model empties infinitely often when $P(A > V) > 0$ is proved, for example, in [Sigman, 1988]). We imagine that at the infinite past $n = -\infty$, we start both (4.7) and (4.1) from empty. We construct the RA model forward in time, while using Lemma 5 for the service times for the FIFO model, so that Lemma 4 applies and we have it in the form of (4.3), for all $t_n \leq 0$ up to and including at time $t_0 = 0$, at which time both models are in stationarity. We might have to continue the construction of the RA model so that W_0 (distributed as π) can be constructed (i.e., enough service times have been initiated by the RA model for using Lemmas 4 and 5). Formally, one can theoretically justify the existence of such infinite from the past versions (that obey Lemma 4) – by using Loynes’ Lemma. Each model (when started empty) satisfies the monotonicity required to use Loynes’ Lemma. In particular, noting that $Q_{RA}(t_n-) = 0$ if and only if $\bar{W}_n = \mathbf{0}$, we conclude that if at any time n it holds that $\bar{W}_n = \mathbf{0}$, then $W_n = \mathbf{0}$. By the Markov property, given that $\bar{W}_n = \mathbf{0} = W_n$, the future is independent of the past for each model, or said differently, *the past is independent of the future*. This remains valid if n is replaced by a stopping time (strong Markov property).

We outline the simulation algorithm steps as follows.

1. Simulate $\{(S_n^{(r)}(i), \bar{W}_{-n}^0(i)) : 0 \leq n \leq N\}, 1 \leq i \leq c\}$ with N as defined in (4.17). If $N = 0$, go to next step. Otherwise, having stored all data, reconstruct \bar{W}_n^0 forward in time from $n = -N$ (initially empty) until $n = 0$, using the recursion (4.14). During this forward-time

reconstruction, re-define V_j as the j -th service initiation used by the RA model (i.e., we are using Lemma 5 to gather service times in the proper order to feed in the FIFO model, which is why we do the re-construction). If at time $n = 0$, there have not yet been N service initiations, then continue simulating the RA model out in forward time until finally there is a N -th service initiation, and then stop. This will require, at most, simulating out to t_n with $n = N^{(+)} = \min\{n \geq 0 : \bar{W}_{-n}^0 = \mathbf{0}\}$. Take the vector $(V_{-N}, V_{-N+1}, \dots, V_{-1})$ and reset $(V_0, V_1, \dots, V_{N-1}) = (V_{-N}, V_{-N+1}, \dots, V_{-1})$. Also, store the interarrival times $(A_{-N}, A_{-N+1}, \dots, A_{-1})$, and reset $(A_0, \dots, A_{N-1}) = (A_{-N}, A_{-N+1}, \dots, A_{-1})$.

2. If $N = 0$, then set $W_0 = \mathbf{0}$ and stop. Otherwise use (4.1) with $W_0 = \mathbf{0}$, recursively go forward in time for N steps until obtaining W_N , by using the N re-set service times $(V_0, V_1, \dots, V_{N-1})$ and interarrival times (A_0, \dots, A_{N-1}) . Reset $W_0 = W_N$.
3. Output W_0 .

Detailed simulation steps are discussed in Appendix B.1. Let τ denote the total number of interarrival times and service times to simulate in order to detect the stopping time N . The following proposition shows that our algorithm will terminate in finite expected time, i.e., $E[\tau] < \infty$. The proof is given in Appendix B.2.

Proposition 4. *If $\rho = \lambda/(c\mu) < 1$, $P(A > V) > 0$, and there exists some $\epsilon > 0$ such that $E[V^{2+\epsilon}] < \infty$, then*

$$E[N] < \infty \quad \text{and} \quad E[\tau] < \infty.$$

4.2.2 Why we can assume that interarrival times are bounded

Lemma 6. *Consider the recursion*

$$D_{n+1} = (D_n + V_n - A_n)^+, \quad n \geq 0, \tag{4.18}$$

where both $\{V_n\}$ and $\{A_n\}$ are non-negative random variables, and $D_0 = 0$.

Suppose for another sequence of non-negative random variables $\{\hat{A}_n\}$, it holds that

$$P(\hat{A}_n \leq A_n, \quad n \geq 0) = 1.$$

Then for the recursion

$$\hat{D}_{n+1} = (\hat{D}_n + V_n - \hat{A}_n)^+, \quad n \geq 0, \quad (4.19)$$

with $\hat{D}_0 = 0$, it holds that

$$P(D_n \leq \hat{D}_n, \quad n \geq 0) = 1. \quad (4.20)$$

Proof. The proof is by induction on $n \geq 0$: Because (w.p.1 in the following arguments) $\hat{A}_0 \leq A_0$, we have

$$D_1 = (V_0 - A_0)^+ \leq (V_0 - \hat{A}_0)^+ = \hat{D}_1.$$

Now suppose the result holds for some $n \geq 0$. Then, $D_n \leq \hat{D}_n$ and by assumption $\hat{A}_n \leq A_n$; hence

$$D_{n+1} = (D_n + V_n - A_n)^+ \leq (\hat{D}_n + V_n - \hat{A}_n)^+ = \hat{D}_{n+1},$$

and the proof is complete. \square

Proposition 5. *Consider the stable RA GI/GI/c model in which $P(A > V) > 0$. In order to use this model to simulate from the corresponding stationary distribution of the FIFO GI/GI/c model as explained in the Section 4.2.1, without loss of generality we can assume that the interarrival times $\{A_n\}$ are bounded: There exists $b > 0$ such that*

$$P(A_n \leq b, \quad n \geq 0) = 1.$$

Proof. By stability, $cE[A] > E[V]$, and by assumption $P(A > V) > 0$. If the $\{A_n\}$ are not bounded, then for $b > 0$, define $\hat{A}_n = \min\{A_n, b\}$, $n \geq 0$; truncated A_n . Choose b sufficiently large so that $cE[\hat{A}] > E[V]$ and $P(\hat{A} > V) > 0$ still holds. Now use the $\{\hat{A}_n\}$ in place of the $\{A_n\}$ to construct an RA model, denoted by \widehat{RA} . Denote this by

$$\hat{W}_n = \left(\hat{W}_n(1), \dots, \hat{W}_n(c) \right),$$

where it satisfies the recursion (4.7) in the form

$$\hat{W}_{n+1} = \left(\hat{W}_n + \tilde{V}_n - \hat{A}_n \mathbf{1} \right)^+, \quad n \geq 0.$$

Starting from $\bar{W}_0 = \hat{W}_0 = \mathbf{0}$, then from Lemma 6, it holds (coordinate-wise) that

$$\bar{W}_n \leq \hat{W}_n, \quad n \geq 0,$$

and thus, if for some $n \geq 0$ it holds that $\hat{W} = \mathbf{0}$, then $\bar{W}_n = \mathbf{0}$ and hence $W_n = \mathbf{0}$ (as explained in our previous section). Since b was chosen ensuring that $cE[\hat{A}] > E[V]$ and $P(\hat{A} > V) > 0$, \hat{W}_n is a stable RA $GI/GI/c$ queue that will indeed empty infinitely often. Thus, we can use it to do the backwards in discrete-time stationary construction until it empties, at time (say) $-\hat{N}$; $\hat{N} = \min\{n \geq 0 : \hat{W}_{-n} = \mathbf{0}\}$. Then, we can re-construct the original RA model (starting empty at time $-\hat{N}$) using the (original untruncated) \hat{N} interarrival times $(A_{-\hat{N}}, A_{-\hat{N}+1}, \dots, A_{-1})$ in lieu of $(\hat{A}_{-\hat{N}}, \hat{A}_{-\hat{N}+1}, \dots, \hat{A}_{-1})$, so as to collect \hat{N} re-ordered V_n needed in construction of W_0 for the target FIFO model. \square

Remark 2. One would expect that the reconstruction of the original RA model in the above proof is unnecessary, that instead we only need to re-construct the \widehat{RA} model until we have \hat{N} service initiations from it, as opposed to \hat{N} service initiations from the original RA model. Although this might be true, the subtle problem is that the order in which service times are initiated in the \widehat{RA} model will typically be different than for the original RA model; they have different arrival processes (counterexamples are easy to construct). Thus, it is not clear how one can utilize Lemma 4 and Lemma 5, and so on. One would need to generalize Lemma 4 to account for truncated arrival times used in the RA model, but not the FIFO model, in perhaps a form such as a variation of Eq. (4.3),

$$P(Q_F(t_n-) \leq Q_{\widehat{RA}}(\hat{t}_n-), \text{ for all } n \geq 0) = 1, \quad (4.21)$$

where $\{\hat{t}_n\}$ is the truncated renewal process. We do not explore this further.

4.2.3 A more efficient algorithm: sandwiching

In this section, we no longer need to assume that $P(A > V) > 0$. (Another method allowing for $P(A > V) = 0$ involving Harris recurrent regeneration is given later in Section 4.6.) Instead of waiting for the workload vector of the $GI/GI/c$ queue under RA discipline to become $\mathbf{0}$, we choose an “inspection time” $t_{-\kappa} < 0$ for some $\kappa \in \mathbb{Z}_+$ to stop the backward simulation of the RA $GI/GI/c$ queue, then construct two bounding processes of the target FIFO $GI/GI/c$ queue and evolve them forward in time, using the same stream of arrivals and service time requirements (in the order of service initiations), until coalescence or time zero. In particular, we let the upper bound process to be a FIFO $GI/GI/c$ queue starting at time $t_{-\kappa}$ with workload vector being $\bar{W}_{-\kappa}^0$, and let the lower

bound process to be a FIFO $GI/GI/c$ queue starting at the same time, $t_{-\kappa}$, from empty, i.e., with workload vector being $\mathbf{0}$.

Let $W(t)$ denote the ordered (ascendingly) workload vector of the original FIFO $GI/GI/c$ queueing process, starting from the infinite past, evaluated at time t . For $t \geq t_{-\kappa}$, we define $W_{-\kappa}^u(t)$ and $W_{-\kappa}^l(t)$ to be the ordered (ascendingly) workload vectors of the upper bound and lower bound processes, initiated at the inspection time $t_{-\kappa}$, evaluated at time t . By our construction and Theorem 3.3 in [Connor and Kendall, 2015],

$$W_{-\kappa}^u(t_{-\kappa}) = \mathcal{S}(\bar{W}_{-\kappa}^0) \geq W(t_{-\kappa}) \geq W_{-\kappa}^l(t_{-\kappa}) = \mathbf{0},$$

and for all $t > t_{-\kappa}$

$$W_{-\kappa}^u(t) \geq W(t) \geq W_{-\kappa}^l(t),$$

where all the above inequalities hold coordinate-wise.

Note that we can evolve the ordered workload vectors of the two bounding processes as follows: For $t_{n-1} \leq t < t_n$ where $-\kappa < n \leq -1$,

$$\begin{aligned} W_{-\kappa}^u(t) &= \mathcal{S}\left(W_{-\kappa}^u(t_{n-1}) + V_{n-1}\mathbf{e}_1 - (t - t_{n-1})\mathbf{1}\right)^+, \\ W_{-\kappa}^l(t) &= \mathcal{S}\left(W_{-\kappa}^l(t_{n-1}) + V_{n-1}\mathbf{e}_1 - (t - t_{n-1})\mathbf{1}\right)^+. \end{aligned} \tag{4.22}$$

Similarly, let $Q(t)$ denote the number of customers in the target FIFO $GI/GI/c$ queueing process (including both waiting in queue and being served), starting from the infinite past, evaluated at time t . For $t \geq t_{-\kappa}$, we let $Q_{-\kappa}^u(t)$ and $Q_{-\kappa}^l(t)$ denote the number of customers (including both waiting in queue and being served) in the upper and lower bound queueing processes, respectively, both initiated at the inspection time $t_{-\kappa}$, evaluated at time t . If at some time $T \in [t_{-\kappa}, 0]$, we observe that $W_{-\kappa}^u(T) = W_{-\kappa}^l(T)$, then it must be true that $W(T) = W_{-\kappa}^u(T) = W_{-\kappa}^l(T)$ and $Q(T) = Q_{-\kappa}^u(T) = Q_{-\kappa}^l(T)$ (because the ordered remaining workload vectors of two bounding processes can only meet when they both have idle servers). We call such time T “coalescence time”, and from then on we have full information of the target FIFO $GI/GI/c$ queue, hence we can continue simulate it forward in time until time 0.

However, if coalescence does not happen by time 0, we can adopt the so-called “binary back-off” method by letting the arrival time $t_{-2\kappa}$ be our new inspection time and redo the above procedure

to detect coalescence. Theorem 3.3 in [Connor and Kendall, 2015] ensures that for any $t_{-\kappa} \leq t \leq 0$

$$W_{-\kappa}^u(t) \geq W_{-2\kappa}^u(t) \geq W(t) \geq W_{-2\kappa}^l(t) \geq W_{-\kappa}^l(t).$$

We summarize the sandwiching algorithm as follows.

1. Simulate $\{(S_n^{(r)}, \bar{W}_{-n}^0 : 0 \leq n \leq \kappa)\}$ with all data stored.
2. Use the stored data to reconstruct \bar{W}_n^0 forward in time from $n = -\kappa$ until $n = 0$, using Eq. (4.14), and re-define V_j as the j^{th} service initiation used by the RA model.
3. Set $W_{-\kappa}^u(t_{-\kappa}) = \mathcal{S}(\bar{W}_{-\kappa}^0)$ and $W_{-\kappa}^l(t_{-\kappa}) = \mathbf{0}$. Use the same stream of interarrival times $(A_{-\kappa}, A_{-\kappa+1}, \dots, A_{-1})$ and service times $(V_{-\kappa}, V_{-\kappa+1}, \dots, V_{-1})$ to simulate $W_{-\kappa}^u(t)$, $W_{-\kappa}^l(t)$ forward in time using Eq. (4.22).
4. If at some time $t \in [t_{-\kappa}, 0]$ we detect $W_{-\kappa}^u(t) = W_{-\kappa}^l(t)$, set $T = t$, $W(T) = W_{-\kappa}^u(T)$, $Q(T) = \sum_{i=1}^c I(W^{(i)}(T) > 0)$, where $W^{(i)}(t)$ is the i -th entry of vector $W(t)$. Then, use the remaining interarrival times and service times to evolve the original FIFO $GI/GI/c$ queue forward in time until time $t_0 = 0$, output $(W(0), Q(0))$ and stop.
5. If no coalescence is detected by time 0, set $\kappa \leftarrow 2\kappa$, then continue to simulate the backward RA $GI/GI/c$ process until $(-\kappa)$ -th arrival, i.e., $\{(S_n^{(r)}, \bar{W}_{-n}^0) : 0 \leq n \leq \kappa\}$, with all data stored. Go to Step 2.

Next we analyze properties of the coalescence time. Define

$$\kappa_-^* = \inf \left\{ n \geq 0 : \inf_{t_{-n} \leq t \leq 0} \|W_{-n}^u(t) - W_{-n}^l(t)\|_\infty = 0 \right\}.$$

If at time $t_{-\kappa_-^*}$ we start an upper bound FIFO $GI/GI/c$ queue with workload vector being $W_{-\kappa_-^*}^u(t_{-\kappa_-^*})$, and a lower bound FIFO $GI/GI/c$ queue with workload vector being $\mathbf{0}$, they will coalesce by time $t_0 = 0$. Therefore, if we simulate the RA system backward in time to $t_{-\kappa_-^*}$, we will be able to detect a coalescence. We next show that $E[-t_{-\kappa_-^*}] < \infty$.

By stationarity, we have that κ_-^* is equal in distribution to

$$\kappa_+^* = \inf \left\{ n \geq 0 : \inf_{0 \leq t \leq t_n} \|W_0^u(t) - W_0^l(t)\|_\infty = 0 \right\},$$

hence $-t_{-\kappa_-^*} \stackrel{d}{=} t_{\kappa_+^*}$.

Proposition 6. *If $\rho = E[V]/(cE[T]) < 1$ and Assumption 1 is in force, then*

$$E \left[t_{\kappa_+^*} \right] < \infty.$$

The proof follows the same argument as in the proof of Proposition 3, so we give a brief proof outline in Appendix B.2.

4.2.4 Continuous-time stationary constructions

For a stable FIFO $GI/GI/1$ queue, let D denote stationary customer delay (time spent in queue (line) waiting); i.e., it has the limiting distribution of $D_{n+1} = (D_n + V_n - A_n)^+$ as $n \rightarrow \infty$.

Independently, let V_e denote a random variable distributed as the *equilibrium distribution* F_e of service time distribution F ,

$$F_e(x) = \mu \int_0^x P(V > y) dy, \quad x \geq 0, \quad (4.23)$$

where $V \sim F$. Let $\nu(t)$ denote total work in system at time t ; the sum of all whole or remaining service times in the system at time t . $D_n = \nu(t_n -)$, and one can construct $\{\nu(t)\}$ via

$$\nu(t) = (D_n + V_n - (t - t_n))^+, \quad t_n \leq t < t_{n+1}.$$

(It is to be continuous from the right with left limits.) Let ν denote stationary workload; i.e., it has the limiting distribution

$$P(\nu \leq x) = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t P(\nu(s) \leq x) ds, \quad x \geq 0. \quad (4.24)$$

The following is well known to hold (see Section 6.3 and 6.4 in [Sigman, 1995], for example):

$$P(\nu > x) = \rho P(D + V_e > x), \quad x \geq 0. \quad (4.25)$$

Letting $H_D(x) = P(D \leq x)$ denote the probability distribution of D , and δ_0 denote the point mass at 0, and $*$ denote convolution of distributions, this means that the distribution of ν can be written as a mixture

$$(1 - \rho)\delta_0 + \rho H_D * F_e.$$

This leads to the following:

Proposition 7. *For a stable ($0 < \rho < 1$) FIFO GI/GI/1 queue, if ρ is explicitly known, and one can exactly simulate from D and F_e , then one can exactly simulate from ν .*

Proof. 1. Simulate a Bernoulli (ρ) r.v. B .

2. If $B = 0$, then set $\nu = 0$. Otherwise, if $B = 1$, then simulate D and independently simulate a copy $V_e \sim F_e$. Set $\nu = D + V_e$. Stop.

□

Another algorithm requiring instead the ability to simulate from G_e (equilibrium distribution of the interarrival time distribution G) instead of F_e follows from another known relation:

$$\nu \stackrel{d}{=} (D + V - A_e)^+, \quad (4.26)$$

where D, V and $A_e \sim G_e$ are independent (see, for example, Eq. (88) on Page 426 in [Wolff, 1989]). Thus by simulating D, V , and A_e , simply set $\nu = (D + V - A_e)^+$. Eq. (4.26) extends analogously to the FIFO GI/GI/ c model, where our objective is to exactly simulate from the time-stationary distribution of the continuous-time Kiefer-Wolfowitz workload vector, $W(t) = (W^{(1)}(t), \dots, W^{(c)}(t))^T$, $t \geq 0$, where it can be constructed via

$$W(t) = \mathcal{S}(W_n + V_n \mathbf{e}_1 - (t - t_n) \mathbf{1})^+, \quad t_n \leq t < t_{n+1}.$$

It is to be continuous from the right with left limits; $W_n = W(t_n^-)$. Total workload $\nu(t)$, for example, is obtained from this via

$$\nu(t) = \sum_{i=1}^c W^{(i)}(t).$$

Let W^* have the time-stationary distribution of $W(t)$ as $t \rightarrow \infty$, let W_0 have the discrete-time stationary distribution π and let V , A_e and W_0 be independent, then

$$W^* \stackrel{d}{=} \mathcal{S}(W_0 + V \mathbf{e}_1 - A_e \mathbf{1})^+. \quad (4.27)$$

So, once we have a copy of W_0 (distributed as π) from our algorithm in Section 4.2.1 or Section 4.2.3, we can easily construct a copy of W^* as long as we can simulate from G_e . Of course, if arrivals are Poisson then the distribution of W^* is identical to that of W_0 by *Poisson Arrivals See Time Averages* (PASTA) property, but otherwise we can use (4.27).

4.3 Numerical experiments

As a sanity check, we have implemented our perfect sampling algorithm in MATLAB for the case of Erlang(k_1, λ)/Erlang(k_2, μ)/ c queue.

Firstly we consider $M/M/c$ queues, which are special cases of Erlang(k_1, λ)/Erlang(k_2, μ)/ c with $k_1 = k_2 = 1$. For the quantity of interest, number of customers in the FIFO $M/M/c$ queue at stationary, we obtain its empirical distribution from a large number of independent runs of our algorithm, and compare it to the theoretical distribution which has a well-established closed form as below:

$$\pi_0 = \left(\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right)^{-1},$$

$$\pi_k = \begin{cases} \pi_0 \cdot (c\rho)^k / k! & \text{if } 0 < k < c \\ \pi_0 \cdot \rho^k c^c / c! & \text{if } k \geq c \end{cases},$$

where $\rho = \lambda/(c\mu) < 1$.

As an example, Figure 4.1 shows the result of such test when $\lambda = 3$, $\mu = 2$ and $c = 2$. Grey bars are the empirical results of 5000 independent draws using our algorithm, and black bars are the theoretical distribution of the number of customers in system from stationarity. A Pearson's chi-squared test between the theoretical and empirical distributions gives a p -value equal to 0.8781, indicating close agreement (i.e., we cannot reject the null hypothesis that there is no difference between these two distributions). For another set of parameters $\lambda = 10$, $\mu = 2$ and $c = 10$, the results are shown in Figure 4.2 with a p -value being 0.6069 for the Pearson's chi-squared fitness test.

For the general Erlang(k_1, λ)/Erlang(k_2, μ)/ c queue when $k_1 > 1$ and $k_2 > 1$ when $\rho = \lambda k_2 / (c\mu k_1) = 0.9$, we compare the empirical distribution of the number of customers in system at stationarity, obtained from a large number of runs of our perfect sampling algorithm, to the numerical results (with precision at least 10^{-4}) provided in Table III of [Hillier and Lo, 1971]. The results for an Erlang(2, 9)/Erlang(2, 5)/ c queue are given in Figure 4.3. Grey bars are the empirical results of 5000 independent draws using our algorithm and black bars are the numerical values given in [Hillier and Lo, 1971]; Again, they are very close to each other. A Pearson's chi-squared test gives a p -value of 0.9464, therefore we cannot reject the null hypothesis that these two distributions

agree well.

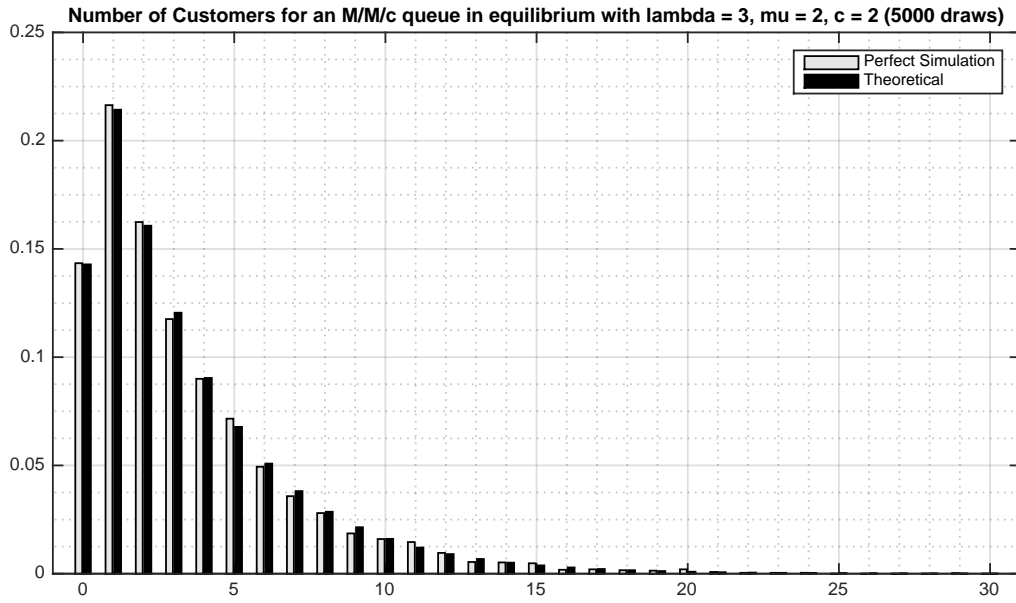


Figure 4.1: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 3$, $\mu = 2$, $c = 2$.

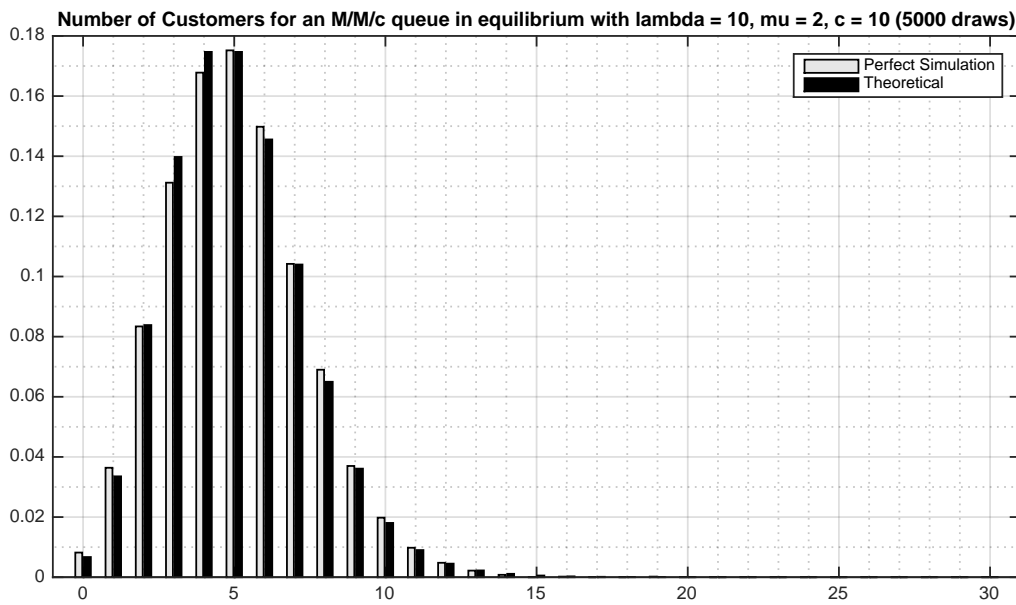


Figure 4.2: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 10$, $\mu = 2$, $c = 10$.

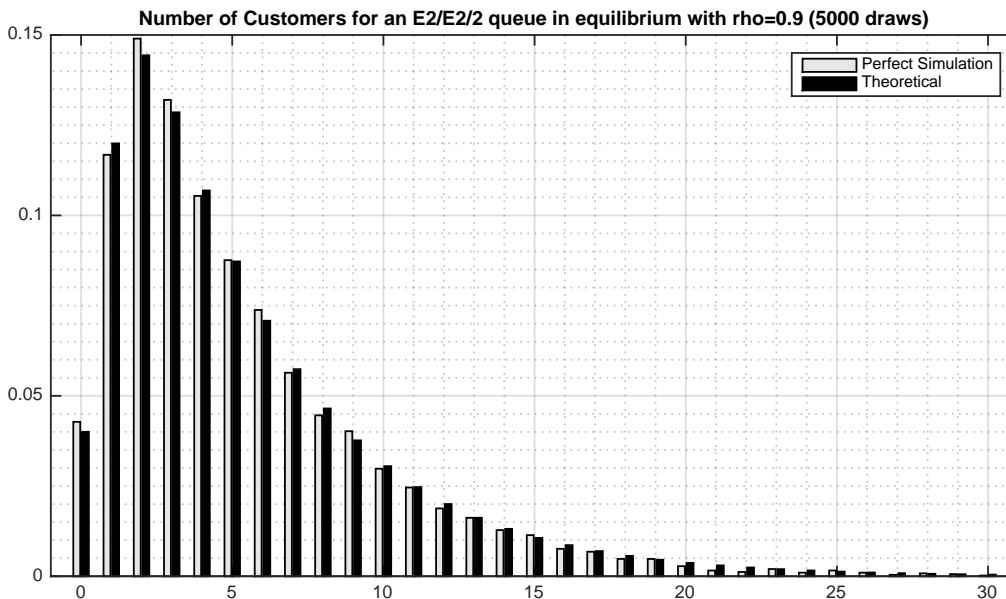


Figure 4.3: Number of customers for an $Erlang(k_1, \lambda)/Erlang(k_2, \mu)/c$ queue in stationarity when $k_1 = 2$, $\lambda = 9$, $k_2 = 2$, $\mu = 5$, $c = 2$ and $\rho = 0.9$.

Next, we run another numerical experiment to compare how far we need to simulate the dominating process backward in time to detect coalescence before (or at) time 0. For the first algorithm given in Section 4.2.1, we let running time $\hat{T} = \sum_{i=1}^N A_{-i}$, i.e., the time taken for the queueing system under RA discipline to become empty the first time; and for the second sandwiching algorithm given in Section 4.2.3, we let running time $\hat{T} = \sum_{i=1}^{\kappa} A_{-i}$, i.e., the time taken for the first successful inspection time in order to detect coalescence before (or at) time 0. In Figure 4.4, we plot the distributions of the time taken for the first time coalescence ever detected under two algorithms, for an $M/M/c$ queue with parameters $\lambda = 10$, $\mu = 2$, $c = 10$, from 5000 runs. The result indicates that the second sandwiching algorithm performs significantly faster than the first one.

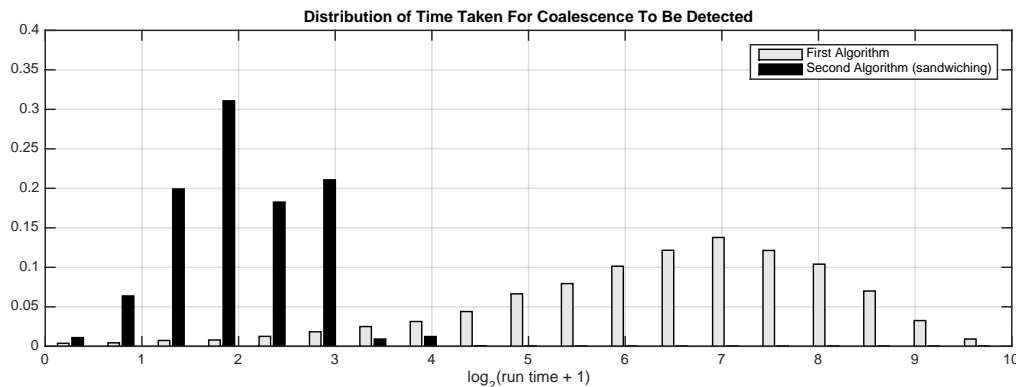


Figure 4.4: Distributions of time taken to detect coalescence under two algorithms for an $M/M/c$ queue

Finally, we test how the computational complexity of the sandwiching algorithm compares to that of the algorithm given in Section 3.3. Notice these two algorithms do look similar: they both use back-off strategies to run two bounding processes from some inspection time and check whether they meet before (or at) time 0. The difference is that in Chapter 3, we use a so-called “vacation system” to construct upper bound process, whereas here we use the same queue but under RA service discipline instead. In the following numerical experiment, we define the computational complexity as the total number of arrivals each algorithm samples backward in time to detect coalescence. Table 4.1 shows how they vary with different values of traffic intensity, ρ , based on 5000 independent runs of both algorithms using the same back-off strategy with same initial $\kappa = 1$. The result suggests that our second sandwiching algorithm outperforms the one proposed in Section 3.3, since the magnitude of the computational complexity does not increase as fast as that of the latter one as traffic intensity increases.

Table 4.1: Simulation result for computational complexities with varying traffic intensities

 $M/M/c$ queue with fixed $\mu = 5$ and $c = 2$

λ	ρ	95% confidence interval of the number of arrivals simulated backwards	
		Algorithm in Section 4.2.3	Algorithm in Section 3.3
5	0.5	54.8194 ± 0.5758	146.5618 ± 2.3598
6	0.6	86.5394 ± 1.0536	308.4448 ± 4.9413
7	0.7	152.6552 ± 2.2695	730.1130 ± 11.2783
8	0.8	337.9544 ± 6.3021	$2,201.8254 \pm 32.1556$
9	0.9	$1,521.3502 \pm 31.8267$	$12,277.8686 \pm 161.5824$

4.4 Infinite server systems and other service disciplines

In this section we sketch how one can utilize our FIFO $GI/GI/c$ results to obtain exact sampling of some other models including the infinite server queue, and the multi-server queue under other service disciplines.

In [Blanchet and Dong, 2013], an exact simulation algorithm is presented for simulating from the stationary distribution of the infinite server queue; the $GI/GI/\infty$ queue. Here we sketch how to utilize our new FIFO $GI/GI/c$ results to accomplish this by using a FIFO $GI/GI/c$ model as an upper bound. The $GI/GI/\infty$ model has an infinite number of servers, there is no line, every arrival enters service immediately upon arrival; the n -th customer arrives at time t_n and departs at time $t_n + V_n$.

For $0 < \lambda/\mu < \infty$, this model is always stable. Let c denote the smallest integer strictly larger than λ/μ ; $c-1 \leq \lambda/\mu < c$. Letting $\nu_\infty(t)$ denote the total amount of work in the $GI/GI/\infty$ model, and $\nu_c(t)$ denote the total amount of work in the (necessarily stable) FIFO $GI/GI/c$ model being fed exactly the same input (of service time requirements and interarrival times), and both starting initially empty, the following is easily established:

$$P(\nu_\infty(t) \leq \nu_c(t), \text{ for all } t \geq 0) = 1, \quad (4.28)$$

hence

$$P(\nu_\infty(t_n-) \leq \nu_c(t_n-), \text{ for all } n \geq 0) = 1. \quad (4.29)$$

(Note that both models use the service times in the same order of initiation, which makes the coupling easy from the start.)

Thus, if, for example $P(A > V) > 0$, then the FIFO model will empty and can be used to detect times when the $GI/GI/\infty$ model will empty. Let $L_\infty(t_n-)$ denote the total number of busy servers in the $GI/GI/\infty$ model as found by C_n .

Simulating the FIFO model backward in time in stationarity (using our previous algorithm), until it first empties, can then be used to detect a time when the $GI/GI/\infty$ model is empty, and then one can construct it back up to time 0 to obtain a stationary copy of $\nu_\infty(t_n-)$ and of $L_\infty(t_n-)$.

Now we consider alternatives disciplines to FIFO for the $GI/GI/c$ model. It is immediate that when service times are generated only when needed by a server, the total number of customers in the system process $\{Q(t)\}$ remains the same under FIFO as under *last-in-first-out* (LIFO) in which the next customer to enter service is the one at the bottom of the line, or *random selection next* (RS) in which the next customer to enter service from the line is selected at random by the server. Thus, they all share the same stationary distribution of $Q(t)$ as $t \rightarrow \infty$, as well as the stationary distribution of $Q(t_n-)$ as $n \rightarrow \infty$. Let Q_0 have this limiting (as $n \rightarrow \infty$) distribution. This fact can be used to exactly simulate, for example, stationary delay D under LIFO or RS (they are not the same as for FIFO). The method (sketch) is as follows: Simulate a copy of Q_0 , jointly with the remaining service times of those in service, by assuming FIFO. This represents the distribution of the system as found in stationarity (at time 0) by arrival C_0 . Consider RS for example. If the line is empty, then define $D_{RS} = 0$; C_0 enters service immediately. Otherwise, place C_0 in the line, and continue simulating but now using RS instead of FIFO. As soon as C_0 enters service, stop and define D_{RS} as that length of time.

4.5 Fork-Join models

The RA recursion (4.7),

$$\bar{W}_{n+1} = \left(\bar{W}_n + \tilde{V}_n - A_n \mathbf{1} \right)^+, \quad n \geq 0, \quad (4.30)$$

is actually a special case for the modeling of *Fork-Join* (FJ) queues (also called *Split and Match*) with c nodes. In an FJ model, each arrival is a “job” with c components, the i -th component requiring service at the i -th FIFO queue. So upon arrival at time t_n , the job splits into its c components to be served. As soon as all c components have completed service, then and only then, does the job depart. Such models are useful in manufacturing applications. The n -th job (C_n) thus arrives with a service time vector attached of the form $\mathbf{V}_n = (V_n(1), \dots, V_n(c))$. Let us assume that the vectors are iid, but otherwise can be generally jointly distributed; for then (4.30) still forms a Markov chain. We will denote this model as the *GI/GI/c – FJ* model. The sojourn time of the i -th component is given by $\bar{W}(i) + V_n(i)$, and thus the sojourn time of the n -th job, C_n , is given by

$$H_n = \max_{1 \leq i \leq c} \{ \bar{W}_n(i) + V_n(i) \}. \quad (4.31)$$

Of great interest is obtaining the limiting distribution of H_n as $n \rightarrow \infty$; we denote a r.v. with this distribution as H^0 . FJ models are notoriously difficult to analyze analytically: Even the special case of Poisson arrivals and iid exponential service times is non-trivial because of the dependency of the c queues through the common arrival process. (A classic paper is [Flatto and Hahn, 1984]). In fact when $c \geq 3$, only bounds and approximations are available. As for exact simulation, there is a paper by Hongsheng Dai [Dai, 2011], in which Poisson arrivals and independent exponential service times are assumed. Because of the continuous-time Markov chain (CTMC) model structure, the author is able to construct (simulate) the time-reversed CTMC to use in a coupling from the past algorithm. But with general renewal arrivals and or general distribution service times, such CTMC methods no longer can be used.

Our simulation method for the RA model outlined in Section 4.2, however yields an exact copy of H^0 for the general *GI/GI/c – FJ* model, under the condition that there exists $\theta > \mathbf{0}$, $\theta \in \mathbb{R}^c$ such that

$$E \left[\exp(\theta^T (\mathbf{V}_1 - A_1 \mathbf{1})) \right] < \infty.$$

First we simulate \bar{W}_0^0 exactly using exponential change of measure method introduced in [Blanchet and Chen, 2015] (we use the same technique for multidimensional simulation in Algorithm 4.2.1),

then simulate a vector of service times $\mathbf{V} = (V(1), \dots, V(c))$ independently and set

$$H^0 = \max_{1 \leq i \leq c} \{\bar{W}_0^0(i) + V(i)\}.$$

Even when the service time components within \mathbf{V} are independent, or the case when service time distributions are assumed to have a finite moment generating function (in a neighborhood of the origin), such results are new and non-trivial.

4.6 The case when $P(A > V) = 0$: Harris recurrent regeneration

For a stable FIFO $GI/GI/c$ queue, the stability condition can be re-written as $E[A_1 + \dots + A_c] > E[V]$, which implies also that $P(A_1 + \dots + A_c > V) > 0$. Thus assuming that $P(A > V) > 0$ is not necessary for stability. When $P(A > V) = 0$, the system will never empty again after starting, and so using consecutive visits to 0 as regeneration points is not possible. But the system does regenerate in a more general way via the use of Harris recurrent Markov chain theory; see [Sigman, 1988] for details and history of this approach. The main idea is that while the system will not empty infinitely often, the number of customers in system process $\{Q_F(t_n-) : n \geq 0\}$ will visit an integer $1 \leq j \leq c - 1$ infinitely often.

For illustration here, we will consider the $c = 2$ case (for the general case $c \geq 2$, the specific regeneration points analogous to what we present here are carefully given in Eq. (4.6) on page 396 of [Sigman, 1988]). Let us assume that $1/2 < \rho < 1$. (Note that if $\rho < 1/2$, then equivalently $E[A] > E[V]$ and so $P(A > V) > 0$; that is why we rule out $\rho < 1/2$ here.) We now assume that $P(A > V) = 0$. This implies that for $\underline{v} \triangleq \inf\{v > 0 : P(V > v) > 0\}$ and $\bar{t} \triangleq \sup\{t > 0 : P(A > t) > 0\}$, we must have $0 < \bar{t} < \underline{v} < \infty$. It is shown in [Sigman, 1988] that for $\epsilon > 0$ sufficiently small, the following event will happen infinitely often (in n) with probability 1,

$$\{Q_{RA}(t_n-) = 1, \bar{W}_n(1) = 0, \bar{W}_n(2) \leq \epsilon, A_n > \epsilon, U_n = 1\}. \quad (4.32)$$

If n is such a time, then at time $n + 1$, we have

$$\{Q_{RA}(t_{n+1}-) = 1, \bar{W}_{n+1}(2) = 0, \bar{W}_{n+1}(1) = (V_n - A_n) \mid A_n > \epsilon\}. \quad (4.33)$$

The point is that C_n finds one server (server 1) empty, and the other queue with only one customer in it, and that customer is in service with a remaining service time $\leq \epsilon$. C_n then enters

service at node 1 with service time V_n ; but since $A_n > \epsilon$, C_{n+1} arrives finding the second queue empty, and the first server has remaining service time $V_n - A_n$ conditional on $A_n > \epsilon$. Under the coupling of Lemma 4, the same will be so for the FIFO model (see Remark 3 below): At such a time n ,

$$\{Q_F(t_n-) = 1, W_n(1) = 0, W_n(2) \leq \epsilon, A_n > \epsilon\}, \quad (4.34)$$

and at time $n + 1$, we have

$$\{Q_F(t_{n+1}-) = 1, W_n(1) = 0, W_n(2) = (V_n - A_n) \mid A_n > \epsilon\}. \quad (4.35)$$

Eqs. (4.33) and (4.35) define positive recurrent regeneration points for the two models (at time $n + 1$); the consecutive times at which regenerations occur forms a (discrete-time) positive recurrent renewal process.

To put this to use, we change the stopping time N given in (4.17) to:

$$\begin{aligned} N + 1 &= \min\{n \geq 1 : Q_{RA}^0(t_{-(n+1)-}) = 1, \bar{W}_{-(n+1)}^0(1) = 0, \\ &\quad \bar{W}_{-(n+1)}^0(2) \leq \epsilon, A_{-(n+1)} > \epsilon, U_{-(n+1)} = 1\}. \end{aligned} \quad (4.36)$$

Then, we do our reconstructions for the algorithm in Section 4.2.1 by starting at time $-N$, with both models starting with the same starting value:

$$\{Q_{RA}(t_{-N}-) = 1, \bar{W}_{-N}^0(2) = 0, \bar{W}_{-N}^0(1) = (V_{-(N+1)} - A_{-(N+1)}) \mid A_{-(N+1)} > \epsilon\} \quad (4.37)$$

and

$$\{Q_F(t_{-N}-) = 1, W_{-N}(1) = 0, W_{-N}(2) = (V_{-(N+1)} - A_{-(N+1)}) \mid A_{-(N+1)} > \epsilon\}. \quad (4.38)$$

Remark 3. The service time used in (4.37) and (4.38) for coupling via Lemma 5, $V_{-(N+1)}$, is in fact identical for both systems because (subtle): At time $-(N + 1)$, both systems have only one customer in system, and thus total work is in fact equal to the remaining service time; so we use Eq. (4.5) to conclude that both remaining service times (even if different) are $\leq \epsilon$ (e.g., that is why (4.34) follow from (4.32)). Meanwhile, $C_{-(N+1)}$ enters service immediately across both systems, so it is indeed the same service time $V_{-(N+1)}$ used for both for this initiation.

Part II

Unbiased Monte Carlo Computations and Applications

Chapter 5

Introduction to Part II

In this part, we propose simple yet powerful techniques that can be used to delete bias that often arise in the implementation of Monte Carlo computations in a wide range of decision making and performance analysis settings, for instance, stochastic optimization and distribution quantile estimation, among others.

There are two key advantages of the estimators that we will present. Firstly, they can be easily implemented in the presence of parallel computing processors, yielding estimates whose accuracy improves as the size of available parallel computing cores increases while keeping the work-per-processor bounded in expectation. Secondly, the confidence intervals can be easily produced in settings in which variance estimators might be difficult to obtain (for example in stochastic optimization problems whose asymptotic variances depend on Hessian information).

To appreciate the advantage of parallel computing with bounded cost per parallel processor, let us consider a typical problem in machine learning applications, which usually involves a sheer amount of data. Because of the technical issues that arise in using the whole data set for training, one needs to resort techniques such as stochastic gradient descent, which is not easy to fully run in parallel, or *sample average approximations* (SAA), which can be parallelized easily but it carries a systematic bias. For both the optimal value function and the optimal policies, we provide estimators that are unbiased, possess finite variance, and can be implemented in finite expected termination time. Thus, our estimators can be directly implemented in parallel, with each parallel processor being assigned an amount of work which is bounded in expectation.

A second example that we shall consider as an application of our techniques arises in steady-

state analysis of stochastic systems. A typical setting of interest is to compute a long-term average of expected cost or reward for running a stochastic system. This problem is classical in the literature of stochastic simulation and it has been studied from multiple angles. Our simple approach provides another way such that the steady-state analysis of regenerative processes can be done without any bias. A key characteristic is that the approach we study involves the same principle underlying the stochastic optimization setting mentioned in the previous paragraph.

Another type of problem that we are able to directly address using our methodology is computing unbiased estimators of distribution quantiles. In addition to being unbiased, all of the estimators have finite work-normalized variance and can be simulated in finite expected termination time, which makes their implementation in parallel computation straightforward.

Applications such as stochastic optimization and quantile estimation allow us to highlight the fact that the variance estimates of our Monte Carlo estimators are straightforward to produce. These variance estimates are important for us to generate asymptotically accurate confidence intervals. In contrast, even though asymptotically unbiased estimators may be available, sometimes these estimators require information about Hessians (as in the optimization setting) or even density information (as in quantile estimation applications) to produce accurate confidence intervals, while our estimators do not require this type of information.

Our estimator relates to the multilevel-Monte Carlo method developed in [Giles, 2008]. We apply the de-biasing techniques introduced in [Rhee and Glynn, 2015] and [McLeish, 2012]. Since the introduction of these techniques, several improvements and applications have been studied, mostly in the context of stochastic differential equations and partial differential equations with random input, see for example [Giles and Szpruch, 2014], [Agarwal and Gobet, 2017], [Khodadadian *et al.*, 2018] and [Crisan *et al.*, 2018].

In [Vihola, 2018], a stratified sampling technique is introduced in order to show that the de-biasing in multi-level Monte Carlo can be achieved virtually at no cost in either asymptotic efficiency or sample complexity relative to the standard (biased) MLMC estimator. The results of [Vihola, 2018] can be applied directly to our estimators in order to improve the variance, but the qualitative rate of convergence (i.e. $O(1/\varepsilon^2)$) remains the same). Another recent paper [Dereich and Mueller-Gronbach, 2017] studies multi-level Monte Carlo in the context of stochastic optimization, but their setting is different from what we consider here and they give a completely different class of

algorithms which are not unbiased.

The rest of this part is organized as follows. In Section 5.1 we discuss the general principle which drives the construction of our unbiased estimators. Then, we apply these principles to the different settings of interest, namely, unbiased estimators for non-linear functions of expectations, stochastic convex optimization and quantile estimation, in Chapter 6. Since the topic of this part is sufficiently different than that of Part I, we will reuse some of the notations that have appeared in the previous part with different meanings.

5.1 The general principles

The general principles are based on the work of [Rhee and Glynn, 2015]. Suppose that one is interested in estimating a quantity of the form $\theta(\mu) \in \mathbb{R}$, where μ is a generic probability distribution, say with support in a subset of \mathbb{R}^d , and $\theta(\cdot)$ is a non-linear map.

A useful example to ground the discussion in the mind of the reader is $\theta(\mu) = g(E_\mu[X])$, where $g: \mathbb{R}^d \rightarrow \mathbb{R}$ is a given function (with regularity properties which will be discussed in the sequel). We use the notation $E_\mu[\cdot]$ to denote the expectation operator under the probability distribution μ . For the sake of simplicity, we will later omit the subindex μ when the context is clear.

We consider the empirical measure μ_n of iid samples $\{X_i \in \mathbb{R}^d : 1 \leq i \leq n\}$, i.e.,

$$\mu_n(dx) = \frac{1}{n} \sum_{i=1}^n \delta_{\{X_i\}}(dx),$$

where $\delta_{X_i}(\cdot)$ is the point mass at X_i for $i = 1, \dots, n$. The sample complexity of producing $\mu_n(\cdot)$ is equal to n . By the strong law of large numbers for empirical measures (Varadarajan's Theorem), $\mu_n \rightarrow \mu$ almost surely in the Prohorov space.

Under mild continuity assumptions, we have that

$$\theta(\mu_n) \rightarrow \theta(\mu) \tag{5.1}$$

as $n \rightarrow \infty$ and often one might expect that $\theta(\mu_n)$ is easy to compute. Then, $\theta(\mu_n)$ becomes a natural and reasonable estimator for $\theta(\mu)$. However, there are several reasons that make it desirable to construct an unbiased estimator with finite variance, say Z , for $\theta(\mu)$; even if $\{\theta(\mu_n)\}_{n \geq 1}$ is asymptotically normal in the sense that $n^{1/2}(\theta(\mu_n) - \theta(\mu)) \Rightarrow \mathcal{N}(0, \sigma_\theta^2)$ with some $\sigma_\theta^2 > 0$. First,

as we mentioned in the introduction, if one copy of Z can be produced in finite expected time, averaging the parallel replications of Z immediately yields an estimate of $\theta(\mu)$, whose accuracy then can be increased by the Central Limit Theorem as the number of parallel replications of Z increases. Second, the variance of Z can be estimated with the natural variance estimator of iid replications of Z , when σ_θ^2 may be difficult to evaluate from the samples (e.g. if $\theta(\mu)$ represents some quantile of μ).

Our goal is to construct a random variable Z such that

$$E[Z] = \theta(\mu), \quad \text{Var}(Z) < \infty,$$

and the expected sample complexity to produce Z is bounded. To serve this goal, we first construct a sequence of random variables $\{\Delta_m : m \geq 0\}$ satisfying the following properties:

Assumption 2. *General assumptions*

(i) *There exists some $\alpha, c \in (0, \infty)$ such that $E[|\Delta_m|^2] \leq c \cdot 2^{-(1+\alpha)m}$,*

(ii) $\sum_{m=0}^{\infty} E[\Delta_m] = \theta(\mu)$,

(iii) *If C_m is the computational cost of producing one copy of Δ_m (measured in terms of sampling complexity), then $E[C_m] \leq c' \cdot 2^m$ for some $c' \in (0, \infty)$.*

If we are able to construct the sequence $\{\Delta_m : m \geq 1\}$ satisfying Assumption 2, then we can construct an unbiased estimator Z as follows. First, sample N from geometric distribution with success parameter r , so that $p(k) = P(N = k) = r(1-r)^k$ for $k \geq 0$. The parameter $r \in (0, 1)$ will be optimized shortly. At this point it suffices to assume that $r \in \left(\frac{1}{2}, 1 - \frac{1}{2^{(1+\alpha)}}\right)$.

Once the distribution of N has been specified, the estimator that we consider takes the form

$$Z = \frac{\Delta_N}{p(N)}, \tag{5.2}$$

where N is independent of the iid sequence $\{\Delta_m\}_{m=0}^{\infty}$. Note that the estimator possess finite variance because $r < 1 - \frac{1}{2^{(1+\alpha)}}$,

$$\begin{aligned} E[Z^2] &= \sum_{k=0}^{\infty} E[Z^2 | N = k] p(k) = \sum_{k=0}^{\infty} E\left[\frac{\Delta_k^2}{p(k)^2} | N = k\right] p(k) \\ &= \sum_{k=0}^{\infty} \frac{E[\Delta_k^2]}{p(k)} \leq c \sum_{k=0}^{\infty} \frac{2^{-(1+\alpha)k}}{p(k)} = \frac{c}{r} \sum_{k=0}^{\infty} \frac{1}{(2^{(1+\alpha)}(1-r))^k} < \infty. \end{aligned} \tag{5.3}$$

Moreover, the unbiasedness of the estimator is ensured by Assumption 2(ii),

$$E[Z] = \sum_{k=0}^{\infty} E[Z | N = k] p(k) = \sum_{k=0}^{\infty} E\left[\frac{\Delta_k}{p(k)}\right] p(k) = \theta(\mu).$$

Finally, because $r > 1/2$, the expected sampling complexity of producing Z , denoted by C , is finite precisely by Assumption 2(iii),

$$E[C] = E[C_N] \leq c' \sum_{k=0}^{\infty} 2^k p(k) = rc' \sum_{k=0}^{\infty} (2(1-r))^k < \infty. \quad (5.4)$$

In [Rhee and Glynn, 2015] the choice of N is optimized in terms of the $E[\Delta_m^2]$ and $E[C_m]$. In [Blanchet and Glynn, 2015] a bound on the work-normalized variance, namely the product

$$\sum_{k=0}^{\infty} \frac{2^{-(1+\alpha)k}}{p(k)} \times \sum_{k=0}^{\infty} 2^k p(k),$$

corresponding to the bounds in the right hand side of (5.3) and (5.4) is optimized, and the resulting optimal choice of $p(k)$'s corresponds to choosing N geometrically distributed with $r = 1 - 2^{-3/2}$ when $\alpha = 1$. Following the same logic, the optimal choice of N should be geometrically distributed with $r = 1 - 2^{-(1+\alpha/2)}$ for the general $\alpha > 0$ case and we advocate this choice for the construction of Z .

The contribution of our work is to study the construction of the Δ_m 's based on the sequence $\{\mu_n : n \geq 1\}$ satisfying Assumption 2 as we now explain. Now our focus is on explaining the high-level ideas at an informal level and provide formal assumptions later for different settings.

Suppose that there exists a function $T_\mu^\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ such that

$$\left. \frac{d}{dt} \theta(\mu + t(\mu_n - \mu)) \right|_{t=0} = \int T_\mu^\theta(x) d(\mu_n - \mu) = E_{\mu_n} [T_\mu^\theta(X)] - E_\mu [T_\mu^\theta(X)].$$

Typically, $T_\mu^\theta(\cdot)$ corresponds to the Riesz representation (if it exists) of the derivative of $\theta(\cdot)$ at μ . Going back to the case in which $\theta(\mu) = g(E_\mu[X])$, assuming that $g(\cdot)$ is differentiable with derivative $Dg(\cdot)$, we have

$$\left. \frac{d}{dt} g(E_\mu[X] + t(E_{\mu_n}[X] - E_\mu[X])) \right|_{t=0} = Dg(E_\mu[X]) \cdot (E_{\mu_n}[X] - E_\mu[X]),$$

so in this setting $T_\mu^\theta(x) = Dg(\int z \mu(dz)) \cdot x = Dg(E_\mu[X]) \cdot x$.

Now, suppose that $\theta(\cdot)$ is smooth in the sense that

$$\theta(\mu) = \theta(\mu_n) + \left(E_\mu [T_\mu^\theta(X)] - E_{\mu_n} [T_\mu^\theta(X)] \right) + \epsilon(n, \mu_n), \quad (5.5)$$

where

$$|\epsilon(n, \mu_n)| = O_p \left(\left| E_\mu [T_\mu^\theta(X)] - E_{\mu_n} [T_\mu^\theta(X)] \right|^2 \right).$$

Basically, the term $\epsilon(n, \mu_n)$ controls the error of the first order Taylor expansion of the map

$$t \mapsto \theta(\mu + t(\mu_n - \mu))$$

around $t = 0$. So, in the context in which $\theta(\mu) = g(E_\mu[X])$, if $g(\cdot)$ is twice continuously differentiable, then we have

$$\begin{aligned} & \epsilon(n, \mu_n) \\ &= \frac{1}{2} (E_{\mu_n}[X] - E_\mu[X])^T \cdot (D^2g)(E_\mu[X]) \cdot (E_{\mu_n}[X] - E_\mu[X]) + o(1), \end{aligned}$$

as $n \rightarrow \infty$.

The key ingredient in the construction of the sequence $\{\Delta_m : m \geq 0\}$ is an assumption of the form

$$\sup_{n \geq 1} n^2 E \left[\left| E_\mu [T_\mu^\theta(X)] - E_{\mu_n} [T_\mu^\theta(X)] \right|^4 \right] < \infty, \quad (5.6)$$

this assumption will typically be followed as a strengthening of a Central Limit Theorem companion to the limit $\mu_n \rightarrow \mu$ as $n \rightarrow \infty$, which would typically yield

$$n^{1/2} \{E_\mu [T_\mu(X)] - E_{\mu_n} [T_\mu(X)]\} \implies W,$$

as $n \rightarrow \infty$ for some W . Under (5.6) the construction of Δ_n satisfying Assumption 2(i) proceeds as follows. Let

$$\mu_{2^n}^E(dx) = \frac{1}{2^n} \sum_{i=1}^{2^n} \delta_{\{X_{2^i}\}}(dx), \quad \mu_{2^n}^O(dx) = \frac{1}{2^n} \sum_{i=1}^{2^n} \delta_{\{X_{2^{i-1}}\}}(dx)$$

and set for $n \geq 1$,

$$\Delta_n = \theta(\mu_{2^{n+1}}) - \frac{1}{2} (\theta(\mu_{2^n}^E) + \theta(\mu_{2^n}^O)). \quad (5.7)$$

The key property behind the construction for Δ_n in (5.7) is that

$$\mu_{2^{n+1}} = \frac{1}{2} (\mu_{2^n}^E + \mu_{2^n}^O),$$

so a linearization of $\theta(\mu)$ will cancel the first order effects implied in approximating μ by $\mu_{2^{n+1}}, \mu_{2^n}^O$ and $\mu_{2^n}^E$. In particular, using (5.5) directly we have that

$$|\Delta_n| \leq |\epsilon(2^{n+1}, \mu_{2^{n+1}})| + |\epsilon(2^n, \mu_{2^n}^O)| + |\epsilon(2^n, \mu_{2^n}^E)|,$$

consequently due to (5.6) we have that

$$E \left[|\Delta_n|^2 \right] = O \left(2^{-2n} \right). \quad (5.8)$$

Once (5.8) is in place, verification of Assumption 2(ii) is straightforward because

$$E [\Delta_n] = E [\theta (\mu_{2^{n+1}})] - E [\theta (\mu_{2^n})],$$

so if we define

$$\Delta_0 = \theta (\mu_2),$$

then

$$\sum_{n=0}^{\infty} E (\Delta_n) = \theta (\mu).$$

Assumption 2(iii) follows directly because the sampling complexity required to produce Δ_m is $C_m = 2^{m+1}$ (assuming each X_i required a unit of sample complexity).

The rest of the paper is dedicated to the analysis of (5.7). The abstract approach described here, in terms of the derivative of $\theta (\mu)$, sometimes is cumbersome to implement under the assumptions that are natural in the applications of interest (for example stochastic optimization). So, we may study the error in (5.7) directly in later applications, but we believe that keeping the high-level intuition described here is useful to convey the generality of the main ideas.

Chapter 6

Unbiased Multi-level Monte Carlo

We study three main application areas of the general principles discussed in Section 5.1. In each setting of interest, we give the unbiased estimator with necessary conditions imposed, then we verify the unbiasedness, finiteness of the variance, and finiteness of the expected computation complexity (corresponding to Assumption 2 in the general principles).

Section 6.1 discusses unbiased estimators for functions of expectations and applications in class steady-state analysis of regenerative processes. Section 6.2 discusses unbiased estimators for both the optimal solution and the optimal objective value of stochastic convex optimization problems, along with applications including linear regression and logistic regression. We also provide some numerical experiment results in this section. Section 6.3 gives an unbiased distribution quantile estimator based on Bahadur representation of sample quantiles.

We will use the order in probability notation $O_p(\cdot)$ for stochastic boundedness; $X_n = O_p(a_n)$ means for every $\epsilon > 0$, there exist finite $M_\epsilon > 0$ and $N_\epsilon > 0$ such that

$$P(\|X_n/a_n\| > M_\epsilon) < \epsilon$$

for all $n \geq N_\epsilon$.

6.1 Non-linear functions of expectations and applications

We first apply the general principle to the canonical example considered in our previous discussion, namely

$$\theta(\mu) = g\left(\int y d\mu(y)\right) = g(E_\mu[X]).$$

Let $\nu = E_\mu[X]$. We will impose natural conditions on $g(\cdot)$ to make sure that the principles discussed in Section 5.1 can be directly applied.

We use $(X_k : k \geq 1)$ to denote an iid sequence of copies of the random variable $X \in \mathbb{R}^d$ from distribution μ . For $k \geq 1$, we define

$$X_k^O = X_{2k-1} \quad \text{and} \quad X_k^E = X_{2k}.$$

Note that the X^O 's correspond to X_k 's indexed by odd values and the X^E 's correspond to the X_k 's indexed by even values. For $k \in \mathbb{N}_+$, let

$$S_k = X_1 + \dots + X_k$$

and similarly let

$$S_k^O = X_1^O + \dots + X_k^O,$$

$$S_k^E = X_1^E + \dots + X_k^E.$$

In this setting, we may define

$$\Delta_n = g\left(\frac{S_{2^{n+1}}}{2^{n+1}}\right) - \frac{1}{2}\left(g\left(\frac{S_{2^n}^O}{2^n}\right) + g\left(\frac{S_{2^n}^E}{2^n}\right)\right)$$

for $n \geq 0$ and let the estimator to be

$$Z = \frac{\Delta_N}{p(N)} + g(X_1), \tag{6.1}$$

where N was defined in Section 5.1.

We now impose precise assumptions on $g(\cdot)$, so that Assumption 2 can be verified for Δ_n . Then we summarize our discussion in Theorem 2 next.

Assumption 3. *Function of expectations assumptions:*

(i) Suppose that $g : \mathbb{R}^d \rightarrow \mathbb{R}$ has linear growth of the form $|g(x)| \leq c_1(1 + \|x\|_2)$ for some $c_1 > 0$, where $\|\cdot\|_2$ denotes the l_2 norm in Euclidian space,

(ii) Suppose g is continuously differentiable in a neighborhood of $\nu = E[X]$, and $Dg(\cdot)$ is locally Holder continuous with exponent $\alpha > 0$, i.e.,

$$\|Dg(x) - Dg(y)\|_2 \leq \kappa(x) \|x - y\|_2^\alpha,$$

where $\kappa(\cdot)$ is bounded on compact sets,

(iii) X has finite $3(1 + \alpha)$ moments, i.e. $E \left[\|X\|_2^{3(1+\alpha)} \right] < \infty$.

Theorem 2. *Suppose that Assumption 3 is forced, then $E[Z] = g(E[X])$, $\text{Var}(Z) < \infty$ and the sampling complexity required to produce Z is bounded in expectation.*

Proof. We first show the unbiasedness of the estimator Z . From Assumption 3(i) we have that

$$|g(S_n/n)|^2 \leq c_1' (1 + \|S_n/n\|_2^2).$$

And because of Assumption 3(iii) we have $E[g(S_n/n)^2] < \infty$, which implies that $\{g(S_n/n) : n \geq 0\}$ is uniformly integrable. For each $n \geq 0$,

$$E[\Delta_n] = E[g(S_{2^{n+1}}/2^{n+1})] - E[g(S_{2^n}/2^n)].$$

With the condition in Assumption 3(ii) that g is continuous in a neighborhood of ν , we derive

$$\begin{aligned} E[Z] &= E \left[\frac{\Delta_N}{p(N)} \right] + E[g(X_1)] = \sum_{n=1}^{\infty} E[\Delta_n] + E[g(X_1)] \\ &= \lim_{n \rightarrow \infty} E[g(S_{2^n}/2^n)] = E \left[\lim_{n \rightarrow \infty} g(S_{2^n}/2^n) \right] = g(E[X]). \end{aligned}$$

Next we show $E[\Delta_n^2] = O(2^{-\min(2, (1+\alpha)n)})$ for all $n \geq 0$. We pick $\delta > 0$ small enough so that $g(\cdot)$ is continuously differentiable in a neighborhood of size δ around ν and the locally Holder continuous condition holds as well.

$$\begin{aligned} |\Delta_n| &= |\Delta_n| I(\max(\|S_{2^n}^O/2^n - \nu\|_2, \|S_{2^n}^E/2^n - \nu\|_2) > \delta/2) \\ &\quad + |\Delta_n| I(\|S_{2^n}^O/2^n - \nu\|_2 \leq \delta/2, \|S_{2^n}^E/2^n - \nu\|_2 \leq \delta/2) \\ &\leq |\Delta_n| I(\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) + |\Delta_n| I(\|S_{2^n}^E/2^n - \nu\|_2 > \delta/2) \\ &\quad + |\Delta_n| I(\|S_{2^n}^O/2^n - \nu\|_2 \leq \delta/2, \|S_{2^n}^E/2^n - \nu\|_2 \leq \delta/2). \end{aligned}$$

When $\|S_{2^n}^O/2^n - \nu\|_2 \leq \delta/2$ and $\|S_{2^n}^E/2^n - \nu\|_2 \leq \delta/2$, we have $\|S_{2^{n+1}}/2^{n+1} - \nu\|_2 \leq \delta$ and $\|S_{2^n}^O/2^n - S_{2^n}^E/2^n\|_2 \leq \delta$, thus

$$\begin{aligned} \Delta_n &= \frac{1}{2} (g(S_{2^{n+1}}/2^{n+1}) - g(S_{2^n}^O/2^n)) + \frac{1}{2} (g(S_{2^{n+1}}/2^{n+1}) - g(S_{2^n}^E/2^n)) \\ &= \frac{1}{4} Dg(\xi_n^O)^T \frac{S_{2^n}^E - S_{2^n}^O}{2^n} + \frac{1}{4} Dg(\xi_n^E)^T \frac{S_{2^n}^O - S_{2^n}^E}{2^n} \\ &= \frac{1}{4} (Dg(\xi_n^O) - Dg(\xi_n^E))^T \frac{S_{2^n}^E - S_{2^n}^O}{2^n}, \end{aligned}$$

where ξ_n^O is some value between $S_{2^n}^O/2^n$ and $S_{2^{n+1}}/2^{n+1}$, and ξ_n^E is some value between $S_{2^n}^E/2^n$ and $S_{2^{n+1}}/2^{n+1}$. It is not hard to see that

$$\|\xi_n^O - \xi_n^E\|_2 = \left\| \frac{U_n^O + U_n^E}{2} \cdot \left(\frac{S_{2^n}^O}{2^n} - \frac{S_{2^n}^E}{2^n} \right) \right\|_2 \leq \left\| \frac{S_{2^n}^E}{2^n} - \frac{S_{2^n}^O}{2^n} \right\|_2.$$

Hence, using the fact that $\kappa(\cdot)$ is bounded on compact sets, we have that there exists a deterministic constant $c \in (0, \infty)$ (depending on δ) such that

$$\begin{aligned} &E \left[|\Delta_n|^2 I(\|S_{2^n}^O/2^n - \nu\|_2 \leq \delta/2, \|S_{2^n}^E/2^n - \nu\|_2 \leq \delta/2) \right] \\ &\leq c E \left[|\kappa(\xi_n^E)|^2 \left\| \frac{S_{2^n}^O - S_{2^n}^E}{2^n} \right\|_2^{2(1+\alpha)} \right] = O_p \left(2^{-(1+\alpha)n} \right) \end{aligned}$$

where the last estimate follows from [Bahr, 1965].

On the other hand we analyze the order of

$$E \left[|\Delta_n|^2 I(\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) \right]. \quad (6.2)$$

If we could assume that the X_i 's have a finite moment generating function in a neighborhood of the origin it would be easy to see that (6.2) decays at a speed which is $o(2^{-(1+\alpha)n})$ (actually the rate would be super-exponentially fast in n). However, we are not assuming the existence of a finite moment generating function, but the existence of finite $3(1+\alpha)$ moments. The intuition that we will exploit is that the large deviations event that is being introduced in (6.2) would be driven (in the worst case) by a large jump (that is, we operate based on intuition borrowed from large deviations theory for heavy-tailed increments). So, following this intuition, we define, for some $\delta' > 0$ small to be determined in the sequel, the set

$$\mathcal{A}_n = \{1 \leq i \leq 2^n : \|X_i^O - \nu\|_2 \geq 2^{n(1-\delta')}\}$$

and $N_n = |\mathcal{A}_n|$. In simple words, N_n is the number of increments defining $S_{2^n}^O$ that are large. Note that

$$\begin{aligned} & E [|\Delta_n|^2 I (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2)] \\ &= E [|\Delta_n|^2 I (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) I (N_n = 0)] \\ & \quad + E [|\Delta_n|^2 I (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) I (N_n \geq 1)]. \end{aligned}$$

We can easily verify using Chernoff bound that for any $\gamma > 0$,

$$P (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2 | N_n = 0) = o(2^{-n\gamma}),$$

which implies that

$$\begin{aligned} & E [|\Delta_n|^2 I (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) I (N_n = 0)] \\ &\leq E [|\Delta_n|^{2(1+\alpha)}]^{1/(1+\alpha)} P (\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2, N_n = 0)^{\alpha/(1+\alpha)} \\ &= o(2^{-n(1+\alpha)}). \end{aligned}$$

On the other hand, note that

$$\begin{aligned} & 2^{-2n} E \left[\|X_i^O - \nu\|_2^2 I (\|X_i^O - \nu\|_2 > 2^{n(1-\delta')(1+\alpha)}) \right] \\ &= 2^{-2n+1} \int_{2^{n(1-\delta')(1+\alpha)}}^{\infty} t P (\|X_i^O - \nu\|_2 > t) dt \\ &\leq 2^{-2n+1} \int_{2^{n(1-\delta')(1+\alpha)}}^{\infty} \frac{E (\|X_i^O - \nu\|_2^{3(1+\alpha)})}{t^{2+3\alpha}} dt \\ &= O(2^{-2n-n(1+3\alpha)(1-\delta')(1+\alpha)}). \end{aligned}$$

Using the previous estimate, it follows easily that

$$E [|\Delta_n|^2 I (N_n = 1)] = O(2^n \cdot 2^{-2n-n(1+3\alpha)(1-\delta')(1+\alpha)}).$$

The previous expression is $O(2^{-2n})$ if $\delta' > 0$ is chosen sufficiently small. Similarly, for any fixed k ,

$$\begin{aligned} & E [|\Delta_n|^2 I (N_n = k)] \\ &= O(2^{(k-2)n} 2^{-n \cdot k(1+3\alpha)(1-\delta')(1+\alpha)}) = O(2^{-2n}). \end{aligned}$$

On the other hand,

$$P(N_n \geq m) = O(2^{nm} 2^{-nm \cdot 3(1-\delta')(1+\alpha)}).$$

We then obtain that by selecting $\delta' > 0$ sufficiently small and m large so that

$$m \cdot \alpha \left(3(1 - \delta') - \frac{1}{1 + \alpha} \right) \geq 2,$$

we conclude

$$E [|\Delta_n|^2 I(N_n \geq m)] \leq E [|\Delta_n|^{2(1+\alpha)}]^{1/(1+\alpha)} P(N_n \geq m)^{\alpha/(1+\alpha)} = O(2^{-2n}).$$

Consequently, we have that

$$\begin{aligned} & E [|\Delta_n|^2 I(\|S_{2^n}^O/2^n - \nu\|_2 > \delta/2) I(N_n \geq 1)] \\ & \leq \sum_{k=1}^{m-1} E [|\Delta_n|^2 I(N_n = k)] + E [|\Delta_n|^2 I(N_n \geq m)] = O(2^{-2n}). \end{aligned}$$

Finally the sampling complexity of producing one copy of Δ_n is

$$C_n = 2^{n+1} + c = O(2^n)$$

with some constant $c > 0$. □

6.1.1 Application to steady-state regenerative simulation

The context of steady-state simulation provides an important instance in which developing unbiased estimators is desirable. Recall that if $(W(n) : n \geq 0)$ is a positive recurrent regenerative process taking values on some space \mathcal{Y} , then for all measurable set A , we have the following limit holds with probability one

$$\pi(A) := \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{n=0}^m I(W(n) \in A) = \frac{E_0 \left[\sum_{n=0}^{\tau-1} I(W(n) \in A) \right]}{E_0[\tau]},$$

where the notation E_0 indicates that $W(\cdot)$ is zero-delayed under the associated probability measure $P_0(\cdot)$. The limiting measure $\pi(\cdot)$ is the unique stationary distribution of the process $W(\cdot)$; for additional discussion on regenerative processes see the appendix on regenerative process in [Asmussen and Glynn, 2007], and also [Asmussen, 2003]. Most ergodic Markov chain that arise in practice are regenerative; certainly all irreducible and positive recurrent countable state-space Markov chains are regenerative.

A canonical example which is useful to keep in mind to conceptualize a regenerative process is the waiting time sequence of the single server queue. In which case, it is well known that the waiting

time of the n -th customer, $W(n)$, satisfies the recursion $W(n+1) = \max(W(n) + Y(n+1), 0)$, where the $Y(n)$'s form an iid sequence of random variables with negative mean. The waiting time sequence regenerates at zero, so if $W(0) = 0$, the waiting time sequence forms a zero-delayed regenerative process. Let $f(\cdot)$ be a bounded measurable function and write

$$X_1 = \sum_{n=1}^{\tau-1} f(W(n)) \quad \text{and} \quad X_2 = \tau,$$

then we can estimate the stationary expectation $E_\pi f(W)$ via the ratio

$$E_\pi [f(W)] = \frac{E_0 [X_1]}{E_0 [X_2]}. \quad (6.3)$$

Since $\tau \geq 1$, it follows that for $g(x_1, x_2) = x_1/x_2$. If $E[|X_1|^{3+\epsilon}] < \infty$ and $E[\tau^{3+\epsilon}] < \infty$ for some $\epsilon > 0$, then assumptions can be verified and Theorem 2 applies.

6.1.2 Additional applications

In addition to steady-state simulation, ratio estimators such (6.3) arise in the context of particle filters and state-dependent importance sampling for Bayesian computations, see [Del Moral, 2004] and [Liu, 2008].

In the context of Bayesian inference, one is interested in estimating expectations from some density $(\pi(y) : y \in \mathcal{Y})$ fo the form $\pi(y) = h(y)/\gamma$, where $h(\cdot)$ is a non-negative function with a given (computable) functional form and $\gamma > 0$ is a normalizing constant which is not computable, but is well defined (i.e. finite) and ensures that $\pi(\cdot)$ is indeed a well defined density on \mathcal{Y} . Since $\gamma > 0$ is unknown one must resort to techniques such as Markov chain Monte Carlo or sequential importance sampling to estimate $E_\pi [f(Y)]$ (for any integrable function $f(\cdot)$), see for instance [Liu, 2008].

Ultimately, the use of sequential importance samplers or particle filters relies on the identity

$$E_\pi [f(Y)] = E_q \left[\frac{h(Y)}{q(Y)} f(Y) \right] / E_q \left[\frac{h(Y)}{q(Y)} \right], \quad (6.4)$$

where $(q(y) : y \in \mathcal{Y})$ is a density on \mathcal{Y} and $E_q[\cdot]$ denotes the expectation operator associated to $q(\cdot)$ (and we use $P_q(\cdot)$ for the associated probability). Of course, we must have that the likelihood ratio $\pi(Y)/q(Y)$ well defined almost surely with respect to $P_q(\cdot)$ and

$$E_q \left[\frac{\pi(Y)}{q(Y)} \right] = 1.$$

Thus, by using sequential importance sampling or particle filters one produces a ratio estimator (6.4) and therefore the application of our result in this setting is very similar to the one described in the previous subsection. The verification of Theorem 2 requires additional assumption on the selection of $q(\cdot)$, which should have heavier tails than $\pi(\cdot)$ in order to satisfy Assumption 3.

6.2 Stochastic convex optimization

In this section, we study a wide range of stochastic optimization problems and we show that the general principle applies. This section studies situations in which, going back to Section 5.1, the derivative T_μ^θ may be difficult to characterize and analyze, but the general principle is still applicable. So, in this section we study its applications directly.

Consider the following constrained stochastic convex optimization problem

$$\begin{aligned} \min \quad & f(\beta) = E_\mu[F(\beta, X)] \\ \text{s.t.} \quad & G(\beta) \leq 0, \end{aligned} \tag{6.5}$$

where $\mathcal{D} = \{\beta \in \mathbb{R}^d : F(\beta) \leq 0\}$ is a nonempty closed subset of \mathbb{R}^d . f is a convex map from \mathbb{R}^d to \mathbb{R} . $G(\beta) = (g_1(\beta), \dots, g_m(\beta))^T$ is a vector-valued convex function for some $m \in \mathbb{N}$. X is a random vector whose probability distribution μ is supported on a set $\Omega \subset \mathbb{R}^k$, and $F : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$.

Let β_* denote the optimal solution and $f_* = f(\beta_*)$ denote the optimal objective value. Lagrangian of problem (6.5) is

$$L(\beta, \lambda) = f(\beta) + \lambda^T G(\beta). \tag{6.6}$$

If $f(\cdot)$ and $g_i(\cdot)$'s are continuously differentiable for $i = 1, \dots, m$, the following *Karush-Kuhn-Tucker* (KKT) conditions are sufficient and necessary for optimality:

$$\nabla_\beta L(\beta_*, \lambda_*) = \nabla f(\beta_*) + \nabla G(\beta_*) \lambda_* = 0, \tag{6.7}$$

$$G(\beta_*) \leq 0, \tag{6.8}$$

$$\lambda_*^T G(\beta_*) = 0, \tag{6.9}$$

$$\lambda_* \geq 0, \tag{6.10}$$

where $\lambda_* \in \mathbb{R}^m$ is the Lagrangian multiplier corresponding to β_* .

One of the standard tools in such settings is the method of Sample Average Approximation (SAA), which consists in replacing the expectations by the empirical means. Suppose we have n

iid copies of the random vector X , denoted as $\{X_1, \dots, X_n\}$, we solve the following optimization problem

$$\begin{aligned} \min \quad & f_n(\beta) = \frac{1}{n} \sum_{i=1}^n F(\beta, X_i) \\ \text{s.t.} \quad & G(\beta) \leq 0 \end{aligned} \tag{6.11}$$

as an approximation to the original problem (6.5). Let β_n denote the optimal solution and let $\hat{f}_n = f_n(\beta_n)$ denote the optimal value of the SAA problem (6.11). The traditional SAA approach is to use them as estimators to the true optimal solution β_* and optimal target value f_* of the problem (6.5). Although the SAA estimators are easy to construct and consistent, they are biased. Proposition 5.6 in [Shapiro *et al.*, 2009] shows $E[\hat{f}_n] \leq f_*$ for any $n \in \mathbb{N}$.

We construct unbiased estimators for the optimal solution and optimal value of problem (6.5) by utilizing the SAA estimators. Let $\beta_{2^{n+1}}, \beta_{2^n}^O, \beta_{2^n}^E$ denote the SAA optimal solutions as

$$\begin{aligned} \beta_{2^{n+1}} &= \arg \min_{G(\beta) \leq 0} f_{2^{n+1}}(\beta) = \arg \min_{G(\beta) \leq 0} \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} F(\beta, X_i), \\ \beta_{2^n}^O &= \arg \min_{G(\beta) \leq 0} f_{2^n}^O(\beta) = \arg \min_{G(\beta) \leq 0} \frac{1}{2^n} \sum_{i=1}^{2^n} F(\beta, X_i^O), \\ \beta_{2^n}^E &= \arg \min_{G(\beta) \leq 0} f_{2^n}^E(\beta) = \arg \min_{G(\beta) \leq 0} \frac{1}{2^n} \sum_{i=1}^{2^n} F(\beta, X_i^E). \end{aligned}$$

Let $\hat{f}_{2^{n+1}} = f_{2^{n+1}}(\beta_{2^{n+1}})$, $\hat{f}_{2^n}^O = f_{2^n}^O(\beta_{2^n}^O)$ and $\hat{f}_{2^n}^E = f_{2^n}^E(\beta_{2^n}^E)$ denote the SAA optimal values. Similarly we let $\lambda_{2^{n+1}}, \lambda_{2^n}^O$ and $\lambda_{2^n}^E$ denote the corresponding Lagrange multipliers.

We define

$$\Delta_n = \hat{f}_{2^{n+1}} - \frac{1}{2} (\hat{f}_{2^n}^O + \hat{f}_{2^n}^E) \quad \text{and} \quad \bar{\Delta}_n = \beta_{2^{n+1}} - \frac{1}{2} (\beta_{2^n}^O + \beta_{2^n}^E)$$

for all $n \geq 0$, then the estimator of the optimal value f_* is

$$Z = \frac{\Delta_N}{p(N)} + \hat{f}_1 \tag{6.12}$$

and the estimator of the optimal solution β_* is

$$\bar{Z} = \frac{\bar{\Delta}_n}{p(N)} + \beta_1, \tag{6.13}$$

where N was defined in Section 5.1. We now impose assumptions in this setting so that Assumption 2 for the general principles can be verified for both Δ_n and $\bar{\Delta}_n$.

Assumption 4. *Stochastic convex optimization assumptions:*

- (i) *The feasible region $\mathcal{D} \subset \mathbb{R}^d$ is compact.*
- (ii) *f has a unique optimal solution $\beta_* \in \mathcal{D}$.*
- (iii) *$F(\cdot, X)$ is finite, convex and twice continuously differentiable on \mathcal{D} a.s.*
- (iv) *There exists a locally bounded measurable function $\kappa : \Omega \rightarrow \mathbb{R}_+$, $\gamma > 0$ and $\delta > 0$ such that*

$$|F(\beta', X) - F(\beta, X)| \leq \kappa(X) \|\beta' - \beta\|^\gamma$$

for all $\beta, \beta' \in \mathcal{D}$ with $\|\beta' - \beta\| \leq \delta$ and $X \in \Omega$; and $\kappa(X)$ has finite moment generating function in a neighborhood of the origin.

- (v) *Define,*

$$M_\beta(t) = E[\exp(t(F(\beta, X_i) - f(\beta)))] \quad (6.14)$$

and assume that there exists $\delta_0 > 0$ and $\sigma^2 > 0$ such that for $|t| \leq \delta_0$,

$$\sup_{\beta \in \mathcal{D}} M_\beta(t) \leq \exp(\sigma^2 t^2 / 2).$$

- (vi) *There is $\delta'_0 > 0$ and $t > 0$ such that*

$$\sup_{\|\beta - \beta_*\| \leq \delta'_0} E[\exp(t \|\nabla_\beta F(\beta, X)\|)] < \infty.$$

- (vii) *$E\left[\left\|\nabla_{\beta\beta}^2 F(\beta_*, X)\right\|^p\right] < \infty$ with some $p > 2$.*

(viii) *$G(\beta) = (g_1(\beta), \dots, g_m(\beta))^T$ and $g_i(\cdot)$ is twice continuously differentiable convex function for all $1 \leq i \leq m$.*

- (ix) *There is $\beta \in \mathcal{D}$ such that $G(\beta) < 0$ (Slater conditions ensures strong duality).*

(x) *LICQ holds at β_* , i.e., the gradient vectors $\{\nabla g_i(\beta_*) : g_i(\beta_*) = 0\}$ are linearly independent (LICQ is the weakest condition to ensure the uniqueness of Lagrangian multiplier; see [Wachsmuth, 2013] for instance).*

- (xi) *Strict complementarity condition holds, i.e., $\lambda_*(i) > 0$ when $g_i(\beta_*) = 0$ for all $i = 1, \dots, m$,*

We summarize the discussion of unbiased estimator for the optimal solution β_* as Theorem 3 in Section 6.2.1, and unbiased estimator for the optimal objective value f_* as Theorem 4 in Section 6.2.2.

6.2.1 Unbiased estimator of optimal solution

In this section, we will utilize the large deviation principles for the SAA optimal solutions develop in [Xu, 2010]. We first provide the following lemma to summarize the LDP.

Lemma 7. *If Assumptions 4(i), 4(ii), 4(iv), 4(v) hold, then for every $\epsilon > 0$, there exist positive constants $c(\epsilon)$ and $\alpha(\epsilon)$, independent of n , such that for n sufficiently large*

$$P(\|\beta_{2^n} - \beta_*\| \geq \epsilon) \leq c_\epsilon \exp(-2^n \alpha(\epsilon)),$$

where $\alpha(\epsilon)$ is locally quadratic at the origin, i.e., $\alpha(\epsilon) = \alpha_0 \epsilon^2$ as $\epsilon \rightarrow 0$ with $\alpha_0 > 0$.

Proof. For $\beta \in \mathcal{D}$, let $I_\beta(z) = \sup_{t \in \mathbb{R}} \{zt - \log M_\beta(t)\}$, where $M_\beta(t)$ is as defined in (6.14). The proof of Theorem 4.1 in [Xu, 2010] has that if the assumptions required in Lemma 7 are all enforced, then

$$\begin{aligned} & P(\|\beta_{2^n} - \beta_*\| \geq \epsilon) \\ & \leq \exp(-2^n \lambda) + \sum_{i=1}^M \exp(-2^n \min(I_{\bar{\beta}_i}(\epsilon/4), I_{\bar{\beta}_i}(-\epsilon/4))), \end{aligned}$$

where $\lambda > 0$, $\{\bar{\beta}_i \in \mathcal{D} : 1 \leq i \leq M\}$ is a v -net constructed by the finite covering theorem, i.e., there exists $v > 0$ such that for every $\beta \in \mathcal{D}$, there exists $\bar{\beta}_i, i \in \{1, \dots, M\}$, $\|\beta - \bar{\beta}_i\| \leq v$,

$$|F(\beta, X) - F(\bar{\beta}_i, X)| \leq \kappa(X) \|\beta - \bar{\beta}_i\|^\gamma \quad \text{and} \quad |f(\beta) - f(\bar{\beta}_i)| \leq \epsilon/4,$$

and

$$\min(I_{\bar{\beta}_i}(\epsilon/4), I_{\bar{\beta}_i}(-\epsilon/4)) \geq \frac{\epsilon^2}{32\sigma^2},$$

by Assumption 4(iv) and Remark 3.1 in [Xu, 2010]. Since the size of v -net, M , grows in polynomial order of ϵ , we complete the proof. \square

Theorem 3. *If Assumptions 4 is in force, then $E[\bar{Z}] = \beta_*$, $\text{Var}(\bar{Z}) < \infty$ and the computation complexity required to produce \bar{Z} is bounded in expectation.*

Proof. If Assumptions 4(i), 4(iii), 4(viii), 4(ix), 4(ii), 4(x) and 4(xi) hold, the following result is given on page 171 of [Shapiro *et al.*, 2009]

$$2^{n/2} \begin{bmatrix} \beta_{2^n} - \beta_* \\ \lambda_{2^n} - \lambda_* \end{bmatrix} \implies \mathcal{N}(0, J^{-1} \Gamma J), \quad (6.15)$$

where

$$J = \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix} \quad \text{and} \quad \Gamma = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix},$$

$H = \nabla_{\beta\beta}^2 L(\beta_*, \lambda_*) \in \mathbb{R}_{d \times d}$, A is the matrix whose columns are formed by vectors $\nabla g_i(\beta_*)$ when $g_i(\beta_*) = 0$ for $i = 1, \dots, m$, and $\Sigma = E \left[(\nabla F(\beta_*, X) - \nabla f(\beta_*)) (\nabla F(\beta_*, X) - \nabla f(\beta_*))^T \right]$. Nonsingularity of J is guaranteed by Assumptions 4(x) and 4(xi).

We first show \bar{Z} is unbiased. For $n \geq 0$,

$$E[\bar{\Delta}_n] = E[\beta_{2^{n+1}}] - E[\beta_{2^n}].$$

Since the feasible region $\mathcal{D} \subset \mathbb{R}^d$ is closed and bounded by Assumption 4(i), $\{\beta_{2^n} : n \geq 0\}$ is uniformly integrable. With $\beta_{2^n} \rightarrow \beta_*$ in (6.15), we have

$$E[\bar{Z}] = \sum_{n=1}^{\infty} E[\bar{\Delta}_n] + E[\beta_1] = \lim_{n \rightarrow \infty} E[\beta_{2^n}] = \beta_*.$$

We next prove $\text{Var}(\bar{Z}) < \infty$ by showing $E[\bar{\Delta}_n \bar{\Delta}_n^T] = O(2^{-(1+\alpha)n})$ with some $\alpha > 0$. Let $m(\beta_*) = E[\nabla_{\beta\beta}^2 F(\beta_*, X)]$. The key ingredients are – firstly we use the large deviation principle (LDP) of $\{\beta_{2^n} : n \geq 0\}$ to get moderate deviation estimates for $\{\beta_{2^n} : n \geq 0\}$, secondly to use extended contraction principle with modified optimization problems to translate the LDP to the sequence of Lagrange multipliers $\{\lambda_{2^n} : n \geq 0\}$. For the first part, we have by Lemma 7 that

$$P(\|\beta_{2^n} - \beta_*\| \geq \epsilon) = \exp(-2^n \alpha(\epsilon) + o(2^n))$$

for all $\epsilon > 0$ sufficiently small and $\alpha(\epsilon) = \alpha_0 \epsilon^2 (1 + o(1))$ as $\epsilon \rightarrow 0$ for $\alpha_0 > 0$. This yields moderate deviation estimates for $\{\beta_{2^n} : n \geq 0\}$. In particular, we let $\epsilon \rightarrow 0$ at a speed of the form $\epsilon = 2^{-\rho n}$ for $1/4 < \rho < 1/2$ and the limit above will still provide the correct rate of convergence, i.e.,

$$P(\|\beta_{2^n} - \beta_*\| \geq 2^{-\rho n}) = \exp\left(-\alpha_0 2^{(1-2\rho)n} + o\left(2^{(1-2\rho)n}\right)\right).$$

Then to translate this LDP to $\{\lambda_{2^n} : n \geq 0\}$, we consider a family of modified optimization problems (indexed by η)

$$\begin{aligned} \min \quad & f_\eta(\beta) = E_\mu[F(\beta, X)] + \eta^T \beta \\ \text{s.t.} \quad & G(\beta) \leq 0 \end{aligned}, \tag{6.16}$$

and its associated optimal solution, $\beta(\eta)$, with the associated Lagrange multiplier, $\lambda(\eta)$, for the modified problem. It follows that $\lambda(\cdot)$ is continuously differentiable as a function of η in a neighborhood of the origin, this is a consequence of Assumptions 4(x) and 4(xi). Both $\beta(\eta)$ and $\lambda(\eta)$ are characterized by the following KKT conditions

$$\nabla f(\beta(\eta)) + \nabla G(\beta(\eta)) \lambda(\eta) = -\eta, \quad (6.17)$$

$$G(\beta(\eta)) \leq 0, \quad (6.18)$$

$$\lambda(\eta)^T G(\beta(\eta)) = 0, \quad (6.19)$$

$$\lambda(\eta) \geq 0. \quad (6.20)$$

By one of the KKT optimality conditions specified in (6.7) for the SAA problem we have that

$$0 = \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_{2^n}, X_i) + \nabla G(\beta_{2^n}) \lambda_{2^n}. \quad (6.21)$$

The previous equality implies that

$$\nabla_{\beta} f(\beta_{2^n}) + \nabla_{\beta} G(\beta_{2^n}) \cdot \lambda_{2^n} = -\bar{\eta}_{2^n},$$

where

$$\bar{\eta}_{2^n} = \frac{1}{2^n} \sum_{i=1}^{2^n} (\nabla_{\beta} F(\beta_{2^n}, X_i) - \nabla_{\beta} f(\beta_{2^n})).$$

Written in this form, we can identify that $\beta_{2^n} = \beta(\bar{\eta}_{2^n})$ and $\lambda_{2^n} = \lambda(\bar{\eta}_{2^n})$. We already know that $\{\beta_{2^n} : n \geq 0\}$ has a large deviations principle, so the LDP can be derived for $\{\bar{\eta}_{2^n} : n \geq 0\}$ by Theorem 2.1 of [Gao and Zhao, 2011] with Assumption 4(vi). Furthermore, the LDP can then be derived for the Lagrange multipliers $\{\lambda_{2^n} = \lambda(\bar{\eta}_{2^n}) : n \geq 0\}$ because $\lambda(\cdot)$ is continuously differentiable as a function of η in a neighborhood of the origin, as we mentioned earlier.

Then, from (6.21) it follows by Taylor expansion that

$$\begin{aligned}
0 &= \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_{2^n}, X_i) + \nabla G(\beta_{2^n}) \lambda_{2^n} \\
&= \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_*, X_i) + \frac{1}{2^n} \sum_{i=1}^{2^n} (\nabla_{\beta} F(\beta_{2^n}, X_i) - \nabla_{\beta} F(\beta_*, X_i)) + \nabla G(\beta_{2^n}) \lambda_{2^n} \\
&= \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_*, X_i) + \nabla G(\beta_*) \lambda_* + \left(\frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta\beta}^2 F(\beta_*, X_i) - m(\beta_*) \right) \cdot (\beta_{2^n} - \beta_*) \\
&\quad + (m(\beta_*) + \lambda_*^T \nabla^2 G(\beta_*)) \cdot (\beta_{2^n} - \beta_*) + \nabla G(\beta_*) (\lambda_{2^n} - \lambda_*) \\
&\quad + \bar{R}_{n,(\beta,\beta)} + \bar{R}_{n,(\lambda,\lambda)} + \bar{R}_{n,(\beta,\lambda)}, \tag{6.22}
\end{aligned}$$

where

$$\begin{aligned}
\bar{R}_{n,(\beta,\beta)} &= O(\|\beta_{2^n} - \beta_*\|^2), \\
\bar{R}_{n,(\lambda,\lambda)} &= O(\|\lambda_{2^n} - \lambda_*\|^2), \\
\bar{R}_{n,(\beta,\lambda)} &= O(\|\beta_{2^n} - \beta_*\| \|\lambda_{2^n} - \lambda_*\|).
\end{aligned}$$

Let

$$\bar{R}_n = \left(\frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta\beta}^2 F(\beta_*, X_i) - m(\beta_*) \right) \cdot (\beta_{2^n} - \beta_*)$$

and let $\Lambda_1 = m(\beta_*) + \lambda_*^T \nabla^2 G(\beta_*) \in \mathbb{R}_{d \times d}$, $\Lambda_2 = \nabla G(\beta_*) \in \mathbb{R}_{d \times m}$. Then we can rewrite (6.22) as

$$\begin{aligned}
&\Lambda_1(\beta_{2^n} - \beta_*) + \Lambda_2(\lambda_{2^n} - \lambda_*) \\
&= - \left(\frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_*, X_i) + \nabla G(\beta_*) \lambda_* + \bar{R}_n + \bar{R}_{n,(\beta,\beta)} + \bar{R}_{n,(\lambda,\lambda)} + \bar{R}_{n,(\beta,\lambda)} \right).
\end{aligned}$$

Note that by Holder's inequality,

$$\begin{aligned}
E[\bar{R}_n \bar{R}_n^T] &\leq E \left[\left\| \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta\beta}^2 F(\beta_*, X_i) - m(\beta_*) \right\|^2 \cdot \|\beta_{2^n} - \beta_*\|^2 \right] \\
&\leq E \left[\left\| \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta\beta}^2 F(\beta_*, X_i) - m(\beta_*) \right\|^p \right]^{2/p} E \left[\|\beta_{2^n} - \beta_*\|^{2p/(p-2)} \right]^{(p-2)/p}, \tag{6.23}
\end{aligned}$$

where

$$E \left[\left\| \frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta\beta}^2 F(\beta_*, X_i) - m(\beta_*) \right\|^p \right] = O(2^{-np/2}) \tag{6.24}$$

by [Bahr, 1965], and by using the moderate large deviation estimate

$$\begin{aligned} & E \left[\|\beta_{2^{n+1}} - \beta_*\|^{2p/(p-2)} \right] \\ = & E \left[\|\beta_{2^{n+1}} - \beta_*\|^{2p/(p-2)} I(\|\beta_{2^{n+1}} - \beta_*\| \geq 2^{-\rho n}) \right] \end{aligned} \quad (6.25)$$

$$\begin{aligned} & + E \left[\|\beta_{2^{n+1}} - \beta_*\|^{2p/(p-2)} I(\|\beta_{2^{n+1}} - \beta_*\| < 2^{-\rho n}) \right] \\ \leq & c' \exp(-\alpha_0(1-2\rho)n) + 2^{-\frac{2p\rho n}{p-2}} = O\left(2^{-\frac{2p\rho n}{p-2}}\right), \end{aligned} \quad (6.26)$$

where $c' > 0$ is some constant. Combining them together we get $E[\bar{R}_n \bar{R}_n^T] = O(2^{-(1+2\rho)n})$.

Similarly we can get

$$\begin{aligned} E \left[\bar{R}_{n,(\beta,\beta)} \bar{R}_{n,(\beta,\beta)}^T \right] &= O(2^{-4\rho n}), \\ E \left[\bar{R}_{n,(\lambda,\lambda)} \bar{R}_{n,(\lambda,\lambda)}^T \right] &= O(2^{-4\rho n}), \\ E \left[\bar{R}_{n,(\beta,\lambda)} \bar{R}_{n,(\beta,\lambda)}^T \right] &= O(2^{-4\rho n}). \end{aligned}$$

Because

$$\begin{aligned} & \Lambda_1 \left(\beta_{2^{n+1}} - \frac{1}{2}(\beta_{2^n}^O + \beta_{2^n}^E) \right) + \Lambda_2 \left(\lambda_{2^{n+1}} - \frac{1}{2}(\lambda_{2^n}^O + \lambda_{2^n}^E) \right) \\ = & \bar{R}_{n+1} - \frac{1}{2}(\bar{R}_n^O + \bar{R}_n^E) + \bar{R}_{n+1,(\beta,\beta)} - \frac{1}{2}(\bar{R}_{n,(\beta,\beta)}^O + \bar{R}_{n,(\beta,\beta)}^E) \\ & + \bar{R}_{n+1,(\lambda,\lambda)} - \frac{1}{2}(\bar{R}_{n,(\lambda,\lambda)}^O + \bar{R}_{n,(\lambda,\lambda)}^E) + \bar{R}_{n+1,(\beta,\lambda)} - \frac{1}{2}(\bar{R}_{n,(\beta,\lambda)}^O + \bar{R}_{n,(\beta,\lambda)}^E), \end{aligned}$$

we have that $E[\bar{\Delta}_n \bar{\Delta}_n^T] = O(2^{-4\rho n})$. Note that $\rho \in (1/4, 1/2)$, it satisfies Assumption 2(i) of the general principles of unbiased estimators in Section 5.1.

The computational cost for producing $\bar{\Delta}_n$, denoted by C_n , is of order $O(2^n)$. After generating 2^{n+1} iid copies of X 's, we can use Newton's method or other root-finding algorithms to solve the KKT condition for optimal solution, or use other classic tools such as subgradient method or interior point method. \square

6.2.2 Unbiased estimator of optimal value

Theorem 4. *If Assumption 4 is in force, then $E[Z] = f_*$, $\text{Var}(Z) < \infty$ and the computation complexity required to produce Z is bounded in expectation.*

Proof. Finite expected computation complexity of producing Δ_n has been discussed in the proof to Theorem 3. We now show the unbiasedness of estimator Z . Since $f_n(\beta) = \frac{1}{n} \sum_{i=1}^n F(\beta, X_i) \rightarrow f(\beta)$,

uniformly on \mathcal{D} , with the result of Proposition 5.2 in [Shapiro *et al.*, 2009] we have $\hat{f}_n \rightarrow f_*$ w.p.1 as $n \rightarrow \infty$. If Assumption 4(v) is in force, $\{\hat{f}_{2^n} : n \geq 0\}$ is uniformly integrable, hence

$$E[Z] = \lim_{n \rightarrow \infty} E[\hat{f}_{2^n}] = f_*.$$

We next prove the estimator Z has finite variance by showing $E[\Delta^2] = O(2^{-4\rho n})$ with $\rho > 1/4$. By Taylor expansion around the unique true optimal solution β_* and the KKT condition,

$$\begin{aligned} \Delta_n &= f_{2^{n+1}}(\beta_{2^{n+1}}) - \frac{1}{2} (f_{2^n}^O(\beta_{2^n}^O) + f_{2^n}^E(\beta_{2^n}^E)) \\ &= \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} F(\beta_{2^{n+1}}, X_i) - \frac{1}{2} \left(\frac{1}{2^n} \sum_{i=1}^{2^n} F(\beta_{2^n}^O, X_i^O) + \frac{1}{2^n} \sum_{i=1}^{2^n} F(\beta_{2^n}^E, X_i^E) \right) \\ &= \left(\frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} \nabla_{\beta} F(\beta_*, X_i) + \nabla G(\beta_*) \lambda_* \right)^T (\beta_{2^{n+1}} - \beta_*) + R_{n+1} \\ &\quad - \frac{1}{2} \left(\frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_*, X_i^O) + \nabla G(\beta_*) \lambda_* \right)^T (\beta_{2^n}^O - \beta_*) + R_n^O \\ &\quad - \frac{1}{2} \left(\frac{1}{2^n} \sum_{i=1}^{2^n} \nabla_{\beta} F(\beta_*, X_i^E) + \nabla G(\beta_*) \lambda_* \right)^T (\beta_{2^n}^E - \beta_*) + R_n^E \\ &\quad - \lambda_*^T \nabla G(\beta_*)^T \left(\beta_{2^{n+1}} - \frac{1}{2} (\beta_{2^n}^O + \beta_{2^n}^E) \right), \end{aligned}$$

where $R_{n+1} = O(\|\beta_{2^{n+1}} - \beta_*\|^2)$, $R_n^O = O(\|\beta_{2^n}^O - \beta_*\|^2)$ and $R_n^E = O(\|\beta_{2^n}^E - \beta_*\|^2)$. By using the moderate LDP explained in the proof of Theorem 3, with $\rho \in (1/4, 1/2)$, we have

$$\begin{aligned} E[R_n^2] &\leq c_1 E[\|\beta_{2^n} - \beta_*\|^4 I(\|\beta_{2^n} - \beta_*\| \geq 2^{-\rho n})] + c_2 E[\|\beta_{2^n} - \beta_*\|^4 I(\|\beta_{2^n} - \beta_*\| < 2^{-\rho n})] \\ &= O(2^{-4\rho n}), \end{aligned}$$

where $c_1 > 0$ and $c_2 > 0$ are some constants. Also similar analysis as (6.23) (6.24) and (6.24) in the proof of Theorem 3 yields

$$E \left[\left(\left(\frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} \nabla_{\beta} F(\beta_*, X_i) + \nabla G(\beta_*) \lambda_* \right)^T (\beta_{2^{n+1}} - \beta_*) \right)^2 \right] = O(2^{-(1+2\rho)n})$$

Combining the fact that $E[\bar{\Delta}_n \bar{\Delta}_n^T] = O(2^{-4\rho n})$, we finally get $E[\Delta_n^2] = O(2^{-4\rho n})$. \square

6.2.3 Applications and numerical examples

6.2.3.1 Linear regression

Linear regression is to solve the following optimization problem

$$\min_{\beta \in \mathbb{R}^{p+1}} MSE = E_{\mu} [F(\beta, (X, y))] = E_{\mu} \left[(y - X^T \beta)^2 \right], \quad (6.27)$$

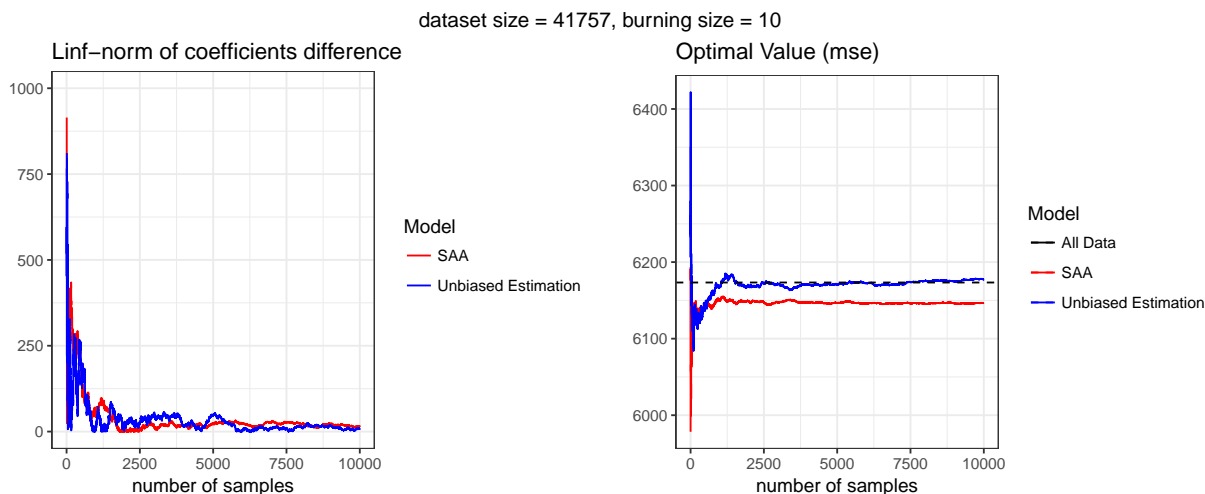
where $X \in \mathbb{R}^{p+1}$ is called independent variables, whose first coordinate is 1, and y is real valued response called dependent variable. The pair (X, y) is from distribution μ . The goal is to find the optimal β_* that minimizes the *mean-squared-error* (MSE).

In many of the real-world problems, we normally have the distribution μ being the empirical measure of all the data available $\{(X_i, y_i) : 1 \leq i \leq n_0\}$, where n_0 denote the total number of data points we have. When n_0 is enormous, it would be difficult and slow to load all the data and do computation at once. Like we mentioned in previous sections, we can take a subsample of the whole dataset to solve the corresponding SAA linear regression, but it results significant estimation bias. With the unbiased estimators presented in Eqs. (6.12) and (6.13), we can take relatively small subsamples and solve them on multiple processors in parallel, without any bias.

We have $F(\beta, (X, y)) = (y - X^T \beta)^2$ strictly convex and twice continuously differentiable in β , so the optimizer is unique. To have all the required conditions listed in Assumption 4 satisfied, we can let $G(\beta) = (g_1(\beta), g_2(\beta), \dots, g_{2(p+1)}(\beta))^T$ with $g_{2i-1}(\beta) = e_i^T \beta - M$ and $g_{2i}(\beta) = -e_i^T \beta - M$ for $1 \leq i \leq p+1$, with $M > 0$ sufficiently large, so that the unique optimizer β_* is in the interior of $\mathcal{D} = \{\beta \in \mathbb{R}^{p+1} : G(\beta) \leq 0\}$. Then, all the conditions follow naturally.

The numerical experiment is to test how the unbiased estimators perform on some real-world dataset. We use Beijing air pollution data (downloaded from the website of UCI machine learning repository), which has 43,824 data points, real-valued PM2.5 concentration and 11 real-valued independent variables including time of a day, temperature, pressure, wind direction and speed, etc. We first use the entire dataset to get the true optimal solution β_* and optimal value f_* as baselines of the experiment. Then, we repeat the SAA approach and our unbiased method for 10,000 times; for both the SAA problem and the unbiased estimation method, we randomly sample a subset of size 2^{N+1} with N geometrically distributed in $\{B, B+1, B+2, \dots\}$. We call such integer value B “burning size”. In Chapter 5.1 we have $B = 0$, which leads to the smallest possible dataset we can get is of size 1. To better control the variance, our experiment uses $B = 10$.

Figure 6.1: Linear regression test on Beijing's PM2.5 data



The left plot of Figure 6.1 has two curves. The red curve shows how $\|\beta_{SAA} - \beta_*\|_\infty$ changes as we increase the number of replications, whereas the blue curve shows the same l_∞ distance between the mean of the unbiased estimators and β_* . At the beginning, both estimators perform volatile, they stabilize as the number of replications gets increased and finally are both close enough to the true optimal solution β_* . The right plot of Figure 6.1 shows how the optimal value estimators from SAA and unbiased estimation method perform as we increase the number of replications. The black dashed horizontal line indicates the level of true optimal value f_* (i.e., the MSE computed by using the entire dataset), the red curve corresponds to the averaged MSE of SAA problems and the blue curve corresponds to the averaged MSE of unbiased estimation method. Clearly, the unbiased estimator outperforms the other as it gets close to f_* after some initial fluctuation, however the SAA estimator gives consistent negative bias, which verifies the theoretic results given in the SAA literature as we mentioned earlier.

6.2.3.2 Logistic regression

Logistic regression is to solve the following optimization problem:

$$\min_{\beta \in \mathbb{R}^n} f(\beta) = E_\mu [F(\beta, (X, y))] = E_\mu [-\log(1 + \exp(-y\beta^T X))], \quad (6.28)$$

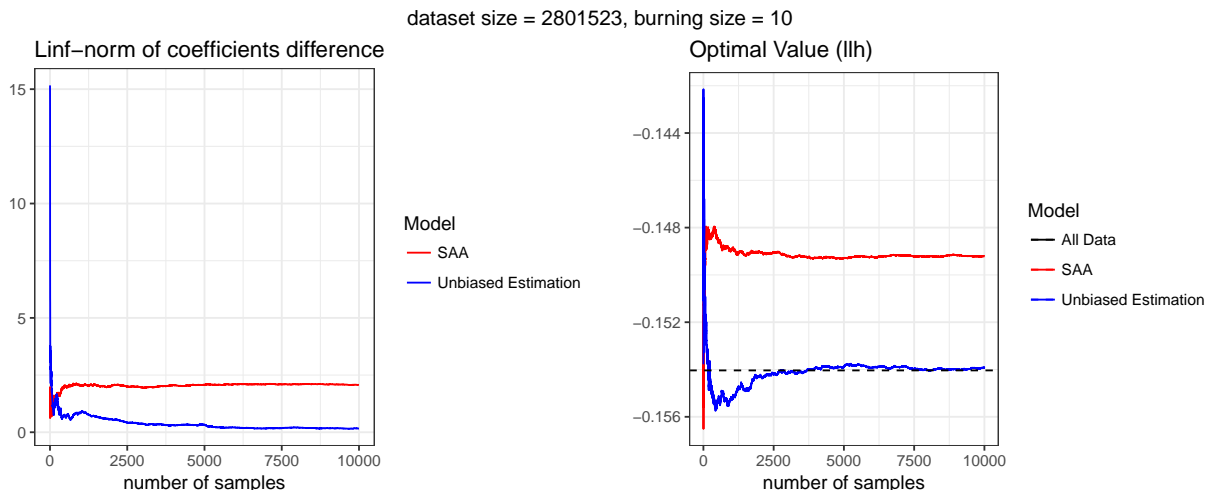
where $X \in \mathbb{R}^{p+1}$ has its first coordinate being 1, and $y \in \{-1, 1\}$ is the label of the class that the data point X falls in. The pair (X, y) is from some distribution μ . The classic logistic regression is to find the optimal coefficient β_* to maximize the log-likelihood, and we give in (6.28) an equivalent problem to minimize the negative of the log-likelihood, i.e., $\min f(\beta)$ with f being strict convex and twice continuous differentiable. Because

$$\nabla_{\beta} F(\beta, (X, y)) = \frac{\exp(-yX^T\beta)}{1 + \exp(-yX^T\beta)} Xy,$$

Assumption 4(iv) is satisfied with $\kappa(X, y) = \|Xy\|_2$ and $\gamma = 1$. To have all the required conditions in Assumption 4 satisfied, we can also let $G(\beta) = (g_1(\beta), \dots, g_{2(p+1)}(\beta))^T$ with $g_{2i-1} = e_i^T \beta - M$ and $g_{2i}(\beta) = -e_i^T \beta - M$ for $1 \leq i \leq p + 1$, with $M > 0$ sufficiently large such that the unique optimizer β_* is in the interior of $\mathcal{D} = \{\beta \in \mathbb{R}^{d+1} : G(\beta) \leq 0\}$.

We run an numerical experiment to check how our unbiased estimators perform, compared to the SAA estimators of both the optimal solution and the optimal objective value. The dataset we use is some online advertising campaign data from Yahoo research, which has 2,801,523 data points, each has 22 real-valued features and one response $y \in \{-1, 1\}$ indicating whether it is a click or not. We first use the entire dataset to get the true optimal solution β_* and optimal value f_* as baselines. Then, for the SAA method and our unbiased estimating method, we run 10000 replications each to see whether they are able to produce a good estimation to β_* and f_* . Again we use the burning size B equal to 10 here.

Figure 6.2: Logistic regression test on AOL’s campaign data



In Figure 6.2, the left plot shows how SAA estimator (in red) and the unbiased estimator (in blue) approach the true optimal solution β_* as we increase the size of replications, and similarly the right plot shows the performance of both estimators for the log-likelihood (i.e., negative of the optimal value f_*), and the baseline level of log-likelihood is represented by the black dashed line. In both cases, our unbiased estimators beat the SAA estimators in terms of unbiasedness.

6.3 Quantile estimation

Suppose $(X_k : k \geq 1)$ are iid with cumulative distribution function $F(x) = \mu((-\infty, x]) = P(X \leq x)$ for $x \in \mathbb{R}$. We define $x_p = x_p(\mu) = \inf\{x \geq 0 : F(x) \geq p\}$ to be the p -quantile of distribution μ for any given $0 < p < 1$. If $F(\cdot)$ is continuous we have that

$$F(x_p) = p.$$

Connecting to the general framework from Section 5.1, here we have $\theta(\mu) := x_p(\mu)$.

We first impose some assumptions.

Assumption 5. *Distributional quantile assumptions:*

- (i) F is at least twice differentiable in some neighborhood of x_p ,
- (ii) $F''(x)$ is bounded in the neighborhood,
- (iii) $F'(x_p) = f(x_p) > 0$,
- (iv) $E[X^2] < \infty$.

Note that Assumptions 5(i), 5(ii) and 5(iii) ensure x_p is the unique p -quantile of distribution μ . By Bahadur representation of sample quantiles in [Bahadur, 1966], we have

$$Y_n = x_p + \frac{np - Z_n}{nf_\mu(x_p)} + R_n, \quad (6.29)$$

where

$$Y_n = (1 - w_n)X_{[np]} + w_n X_{[np]+1}, \quad w_n = np - [np] \in [0, 1), \quad (6.30)$$

i.e., the sample p -quantile of sample (X_1, \dots, X_n) , $Z_n = \sum_{i=1}^n I(X_i \leq x_p)$ and $R_n = O(n^{-3/4} \log n)$ as $n \rightarrow \infty$ almost surely.

Lemma 8. *If Assumption 5 is in force, $\sup_{n \geq 1/p} E[Y_n^2] < \infty$.*

Proof. Just follow Bahadur's proof. Let

$$G_n(x, \omega) = (F_n(x, \omega) - F_n(x_p, \omega)) - (F(x) - F(x_p)),$$

and let I_n be an open interval $(x_p - a_n, x_p + a_n)$ with the constant $a_n \sim \log n / \sqrt{n}$ as $n \rightarrow \infty$. Define

$$H_n(\omega) = \sup \{|G_n(x, \omega)| : x \in I_n\}.$$

By Lemma 1 in [Bahadur, 1966], $H_n(\omega) \leq K_n(\omega) + \beta_n$ with $\beta_n = O(n^{-3/4} \log n)$, $\sum_n \mathbb{P}(K_n \geq \gamma_n) < \infty$ and $\gamma_n = cn^{-3/4} \log n$. By Lemma 2 in [Bahadur, 1966] we have $Y_n \in I_n$ for sufficiently large n w.p.1. Let $n_* = \sup_n \{K_n \geq \gamma_n \text{ or } Y_n \notin I_n\} < \infty$, then for all $n \geq 1/p$

$$E[Y_n^2] = E[Y_n^2 I(n \leq n_*)] + E[Y_n^2 I(n > n_*)],$$

where

$$E[Y_n^2 I(n \leq n_*)] \leq E\left[\sum_{i=1}^n X_i^2 I(n \leq n_*)\right] \leq n_* E[X^2] < \infty,$$

and

$$\begin{aligned} E[Y_n^2 I(n > n_*)] &= E\left[\left(x_p + \frac{np - Z_n}{nf(x_p)} + R_n\right)^2 I(n > n_*)\right] \\ &\leq 3x_p^2 + \frac{3pq}{nf(x_p)^2} + 3E[R_n^2 I(n > n_*)] \\ &\leq 3x_p^2 + \frac{3pq}{nf(x_p)^2} + 3E[H_n^2 I(n > n_*)] \\ &\leq 3x_p^2 + \frac{3pq}{nf(x_p)^2} + 6\gamma_n^2 + 6\beta_n^2 < \infty. \end{aligned}$$

Combining these two parts together we can conclude $\sup_n E[Y_n^2] < \infty$. \square

We let Y_{2n+1} denote the sample p -quantile of (X_1, \dots, X_{2n+1}) , let Y_{2n}^O denote the sample p -quantile of the odd indexed sub-sample (X_1^O, \dots, X_{2n}^O) and let Y_{2n}^E denote the sample p -quantile of the even indexed sub-sample (X_1^E, \dots, X_{2n}^E) . Then, define

$$\Delta_n = Y_{2n+1} - \frac{1}{2}(Y_{2n}^O + Y_{2n}^E). \quad (6.31)$$

Let $n_b = \min\{n \in \mathbb{N} : n \geq 1/p\}$. We let the geometrically distributed random variable N to take values on $\{n_b, n_b + 1, \dots\}$ with $p(n) = P(N = n) > 0$ for all $n \geq n_b$. Define the estimator to be

$$Z = \frac{\Delta_N}{p(N)} + Y_{2n_b}. \quad (6.32)$$

Theorem 5. *If Assumption 5 are in force, then $E[Z] = x_p$, $\text{Var}(Z) < \infty$ and the computation complexity required to produce Z is bounded in expectation.*

Proof. We first show the unbiasedness of Z . Uniform integrability of $\{Y_{2^n} : n \geq n_b\}$ is established in Lemma 8 with Assumption 5(iv) holds true, so we have

$$E[Z] = \sum_{n=n_0}^{\infty} E[\Delta_n] + E[Y_{2^{n_0}}] = \lim_{n \rightarrow \infty} E[Y_{2^n}] = E\left[\lim_{n \rightarrow \infty} Y_{2^n}\right] = x_p.$$

We next show $\text{Var}(Z) < \infty$. With (6.29) we have

$$\begin{aligned} \Delta_n &= \left(x_p + \frac{2^{n+1}p - Z_{2^{n+1}}}{2^{n+1}f(x_p)} + R_{2^{n+1}}\right) - \frac{1}{2} \left[\left(x_p + \frac{2^n p - Z_{2^n}^O}{2^n f(x_p)} + R_{2^n}^O\right) + \left(x_p + \frac{2^n p - Z_{2^n}^E}{2^n f(x_p)} + R_{2^n}^E\right)\right] \\ &= R_{2^{n+1}} - \frac{1}{2} (R_{2^n}^O + R_{2^n}^E) \\ &= O\left(n \cdot 2^{-3n/4}\right) \quad w.p.1, \end{aligned}$$

thus $\Delta_n^2 = O(n^2 \cdot 2^{-3n/2})$. Again by Lemma 8 and (6.29), we have $\sup_n E[R_n^2] < \infty$, hence $\{\Delta_n : n \geq n_0\}$ is uniformly integrable and $E[\Delta_n^2] = O(n^2 \cdot 2^{-3n/2})$. If we choose $p(n) = r(1-r)^{n-n_b}$ with $r < 1 - \frac{1}{2\sqrt{2}}$ for $n \geq n_b$, then

$$E\left[\left|\frac{\Delta_N}{p(N)}\right|^2\right] = \sum_{n=n_b}^{\infty} \frac{E[\Delta_n^2]}{p(n)} < \infty,$$

thus $\text{Var}(Z) < \infty$.

Finally we show the computation cost of generating Δ_n is finite in expectation. Each replication of Z involves simulating 2^{N+1} independent copies of X . If we adopt the selection method based on random partition introduced in [Blum *et al.*, 1973], then it will cost us $O(2^{N+1})$ time to identify the sample p -quantiles $Y_{2^{N+1}}$, $Y_{2^N}^O$, and $Y_{2^N}^E$. Therefore by letting N be an independent geometrically distributed random variable with success parameter $r \in (1/2, 1 - 2^{-3/2})$, Z is an unbiased estimator of the true unique p -quantile x_p and it has finite work-normalized variance. \square

Part III

Bibliography

Bibliography

- [Agarwal and Gobet, 2017] A. Agarwal and E. Gobet. Finite variance unbiased estimation of stochastic differential equations. In *2017 Winter Simulation Conference (WSC)*, pages 1950–1961, Dec 2017.
- [Asmussen and Glynn, 2007] S. Asmussen and P.W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer-Verlag New York, 2007.
- [Asmussen *et al.*, 1992] S. Asmussen, P.W. Glynn, and H. Thorisson. Stationarity detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2(2):130–157, 1992.
- [Asmussen, 2003] S. Asmussen. *Applied Probability and Queues*. Springer, 2 edition, 2003.
- [Bahadur, 1966] R.R. Bahadur. A note on quantiles in large samples. *The Annals of Mathematical Statistics*, 37(3):577–580, 06 1966.
- [Bahr, 1965] B.V. Bahr. On the convergence of moments in the Central Limit Theorem. *The Annals of Mathematical Statistics*, 36(3):808–818, 1965.
- [Blanchet and Chen, 2015] J. Blanchet and X. Chen. Steady-state simulation of reflected Brownian motion and related stochastic networks. *Annals of Applied Probability*, 25(6):3209–3250, 2015.
- [Blanchet and Dong, 2013] J. Blanchet and J. Dong. Perfect sampling for infinite server and loss systems. *Advances in Applied Probability*, 47, 12 2013. Forthcoming in *Advances in Applied Probability*.

- [Blanchet and Glynn, 2015] J. Blanchet and P.W. Glynn. Unbiased Monte Carlo for optimization and functions of expectations via multi-level randomization. In *Proceedings of the 2015 Winter Simulation Conference*, WSC '15, pages 3656–3667. IEEE Press, 2015.
- [Blanchet and Sigman, 2011] J. Blanchet and K. Sigman. On exact sampling of stochastic perpetuities. *Journal of Applied Probability*, 48(A):165–182, 2011.
- [Blanchet and Wallwater, 2015] J. Blanchet and A. Wallwater. Exact sampling for the stationary and time-reversed queues. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 25(4):26:1–26:27, 2015.
- [Blum *et al.*, 1973] M. Blum, R.W. Floyd, V. Pratt, R.L. Rivest, and R.E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448 – 461, 1973.
- [Chen and Yao, 2013] H. Chen and D.D. Yao. *Fundamentals of queueing networks: Performance, asymptotics, and optimization*, volume 46. Springer Science & Business Media, 2013.
- [Connor and Kendall, 2007] S.B. Connor and W.S. Kendall. Perfect simulation for a class of positive recurrent Markov chains. *The Annals of Applied Probability*, 17(3):781–808, 06 2007.
- [Connor and Kendall, 2015] S.B. Connor and W.S. Kendall. Perfect simulation of M/G/c queues. *Advances in Applied Probability*, 47(4):1039–1063, 12 2015.
- [Corcoran and Tweedie, 2001] J.N. Corcoran and R.L. Tweedie. Perfect sampling of ergodic Harris chains. *The Annals of Applied Probability*, 11(2):438–451, 05 2001.
- [Crisan *et al.*, 2018] D. Crisan, P. Del Moral, J. Houssineau, and A. Jasra. Unbiased multi-index Monte Carlo. *Stochastic Analysis and Applications*, 36(2):257–273, 2018.
- [Dai, 2011] H. Dai. Exact Monte Carlo simulation for Fork-Join networks. *Advances in Applied Probability*, 43(2):484–503, 03 2011.
- [Del Moral, 2004] P. Del Moral. *Feynman-Kac Formulae Genealogical and Interacting Particle Systems with Applications*. Springer-Verlag, 2004.
- [Dereich and Mueller-Gronbach, 2017] S. Dereich and T. Mueller-Gronbach. General multilevel adaptations for stochastic approximation algorithms. *arXiv:1506.05482*, 2017.

- [Ensor and Glynn, 2000] K. Ensor and P.W. Glynn. Simulating the maximum of a random walk. *Journal of Statistical Planning and Inference*, 85:127–135, 2000.
- [Flatto and Hahn, 1984] L. Flatto and S. Hahn. Two parallel queues created by arrivals with two demands. *SIAM Journal on Applied Mathematics*, 44(5):1041–1053, 1984.
- [Foss and Chernova, 2001] S.G. Foss and N.I. Chernova. On optimality of the FCFS discipline in multiserver queueing systems and networks. *Siberian Mathematical Journal*, 42(2):372–385, 2001.
- [Foss and Konstantopoulos, 2006] S.G. Foss and T. Konstantopoulos. Lyapunov function methods. Lecture Notes, 2006.
- [Foss and Tweedie, 1998] R.L. Foss and R.L. Tweedie. Perfect simulation and backward coupling. *Stochastic Models*, 14:187–203, 1998.
- [Foss, 1980] S.G. Foss. Approximation of multichannel service systems. *Sibirsk. Mat. Zh.*, 21(6):132–140, 1980.
- [Gao and Zhao, 2011] F. Gao and X. Zhao. Delta method in large deviations and moderate deviations for estimators. *The Annals of Statistics*, 39(2):1211–1240, 2011.
- [Garmarnik and Goldberg, 2013] D. Garmarnik and D. Goldberg. Steady-state GI/GI/n queue in the Halfin-Whitt regime. *Annals of Applied Probability*, 23:2382–2419, 2013.
- [Giles and Szpruch, 2014] M.B. Giles and L. Szpruch. Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation. *The Annals of Applied Probability*, 24(4):1585–1620, 08 2014.
- [Giles, 2008] M.B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [Hillier and Lo, 1971] F.S. Hillier and F.D. Lo. Tables for multiple-server queueing systems involving Erlang distributions. Technical Report 31, Department of Operations Research, Stanford University, 1971.
- [Kelly, 1979] F.P. Kelly. *Reversibility and stochastic networks*, volume 40. Wiley, 1979.

- [Kendall and Møller, 2000] 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*, pages 844–865, 2000.
- [Kendall, 1998] W.S. Kendall. Perfect simulation for the area-interaction point process. In *Probability towards 2000*, pages 218–234. Springer, 1998.
- [Kendall, 2004] W.S. Kendall. Geometric ergodicity and perfect simulation. *Electron. Comm. Probab.*, 9:140–151, 2004.
- [Khodadadian *et al.*, 2018] A. Khodadadian, L. Taghizadeh, and C. Heitzinger. Optimal multi-level randomized quasi-Monte-Carlo method for the stochastic drift–diffusion-Poisson system. *Computer Methods in Applied Mechanics and Engineering*, 329:480 – 497, 2018.
- [Liu *et al.*, 1995] Z. Liu, P. Nain, and D. Towsley. Sample path methods in the control of queues. *Queueing Systems*, 21(1-2):293–335, 1995.
- [Liu, 2008] J.S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2008.
- [McLeish, 2012] D. McLeish. A general method for debiasing a Monte Carlo estimator. *Monte Carlo Methods and Applications*, 17(4):301–315, 2012.
- [Propp and Wilson, 1996] J. Propp and D. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.
- [Rhee and Glynn, 2015] C. Rhee and P.W. Glynn. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.
- [Rubinstein and Kroese, 2011] R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.
- [Shapiro *et al.*, 2009] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 2009.
- [Sigman, 1988] K. Sigman. Regeneration in tandem queues with multiserver stations. *Journal of Applied Probability*, 25(2):291—403, 1988.

- [Sigman, 1995] K. Sigman. *Stationary Marked Point Processes. An Intuitive Approach*. Chapman & Hall, New York, 1995.
- [Sigman, 2011] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue. *Journal of Applied Probability*, 48A:209–216, 2011.
- [Sigman, 2012] K. Sigman. Exact sampling of the stationary distribution of the FIFO M/G/c queue: the general case for $\rho < c$. *Queueing Systems*, 70:37–43, 2012.
- [Vihola, 2018] M. Vihola. Unbiased estimators and multilevel Monte Carlo. *Operations Research*, 66(2):448–462, 2018.
- [Wachsmuth, 2013] G. Wachsmuth. On LICQ and the uniqueness of Lagrange multipliers. *Operations Research Letters*, 41(1):78–80, 2013.
- [Wolff, 1977] R.W. Wolff. An upper bound for multi-channel queues. *Journal of Applied Probability*, 14:884–888, 1977.
- [Wolff, 1987] R.W. Wolff. Upper bounds on work in system for multichannel queues. *Journal of Applied Probability*, 24(2):547–551, 1987.
- [Wolff, 1989] R.W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, New Jersey, 1989.
- [Xu, 2010] H. Xu. Uniform exponential convergence of sample average random functions under general sampling with applications in stochastic programming. *Journal of Mathematical Analysis and Applications*, 368, 2010.

Part IV

Appendices

Appendix A

Appendix to Chapter 3

A.1 The iid property of the coupled service times and independence of the arrival process

In order to explain why the V_n form an iid sequence, independent of the sequence $\mathcal{T}^0 = \{T_n^0 : n \in \mathbb{Z} \setminus \{0\}\}$, it is useful to keep in mind the diagram depicted in Figure A.1, which illustrates a case involving two servers, $c = 2$.

The assignment of the service times, as we shall explain, can be thought of as a procedure similar to a Tetris game. Arrival times are depicted by dotted horizontal lines which go from left to right, starting at the left most vertical line, which is labeled “Arrivals”. Think of the time line going, vertically, from the bottom of the graph (past) to the top of the graph (future).

In the right-most column in Figure A.1, we indicate the queue length, right at the time of a depicted arrival (and thus, including the arrival itself). So, for example, the first arrival depicted in Figure A.1 observes one customer waiting and thus, including the arrival himself, there are two customers waiting in queue.

The Tetris configuration observed by an arrival at time T is comprised of two parts: (i) the receding horizon, which corresponds to the remaining incomplete blocks, and (ii) the landscape, comprised of the configuration of complete blocks. So, for example, the first arrival in Figure A.1 observes a receding horizon corresponding to the two white remaining blocks, which start from the dotted line at the bottom. The landscape can be parameterized by a sequence of block sizes, and

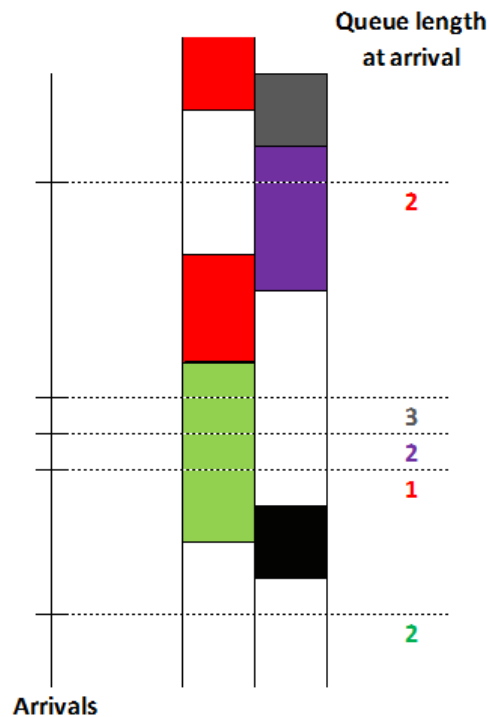


Figure A.1: Matching procedure of service times to arrival process

the order of the sequence is given by the way in which the complete blocks appear from bottom to top – this is precisely the Tetris-game assignment. There are no ties because of the continuous time stationarity and independence of the underlying renewal processes. The colors are, for the moment, not part of the landscape. We will explain the meaning of the colors momentarily.

The assignment of the service times is done as follows: The arriving customer reads off the right-most column (with heading “Queue length at arrival”) and selects the block size labeled precisely with the number indicated by the “Queue length at arrival”. So, there are two distinctive quantities to keep in mind assigned to each player (i.e., arriving customer): (a) the landscape (or landscape sequence, which, as indicated, can be used to reconstruct the landscape), and (b) the *service time*, which is the complete block size occupying the “Queue length at arrival”-th position in the landscape sequence.

The color code in Figure A.1 simply illustrates quantity (b) for each of the arrivals. So, for example, the first arrival, who reads “Queue length at arrival = 2” (which we have written in green color), gets assigned the second complete block, which we have depicted in green. Similarly, the

second arrival depicted reads off the number “1” (written in red) and gets assigned the first red block depicted (from bottom to top). The very first complete block (from bottom to top), which is depicted in black, corresponds to the service time assigned to the customer ahead of the customer who collected the green block. The number “1” (in red) is obtained by observing that the customer with the initial black block has departed.

Now we argue the following properties:

- (1) The service times are iid copies of V .
- (2) The service times are independent of \mathcal{T}^0 .

About property (1): The player arriving at time T reads a number, corresponding to the queue length, which is obtained by the *past filtration* \mathcal{F}_T generated by $\cup_{k \in \mathbb{Z} \setminus \{0\}, 0 \leq i \leq c} \{T_k^i : T_k^i \leq T\}$. Conditional on the receding horizon (i.e., remaining incomplete block sizes), \mathcal{R}_T , the past filtration is independent of the landscape. This is simply the Markov property applied to the forward residual lifetime process of each of the c renewal processes represented by the c middle columns. Moreover, conditional on \mathcal{R}_T , each landscape forms a sequence of iid copies of V because of the structure of the underlying c renewal processes corresponding to the middle columns. So, let $Q(T)$ denote the queue length at time T (including the arrival at time T), which is a function of the past filtration, and let $\{L_T(k) : k \geq 1\}$ be the landscape sequence observed at time T , so that $L_T(Q(T))$ is the service time of the customer who arrives at time T . We then have that for any positive and bounded continuous function $f(\cdot)$,

$$E[f(L_T(Q(T))) | \mathcal{R}_T] = E[f(L_T(1)) | \mathcal{R}_T] = E[f(V)],$$

precisely because, conditional on \mathcal{R}_T , $Q(T)$ (being \mathcal{F}_T measurable) is independent of $\{L_T(k) : k \geq 1\}$.

To verify the iid property, let f_1, f_2 be non-negative and bounded continuous functions. Assume that $T_1 < T_2$ are arrival times in \mathcal{T}^0 (not necessarily consecutive). Then,

$$\begin{aligned} & E[f_1(L_{T_1}(Q(T_1))) f_2(L_{T_2}(Q(T_2)))] \\ &= E[E[f_1(L_{T_1}(Q(T_1))) f_2(L_{T_2}(Q(T_2)))] | \mathcal{F}_{T_2}, \mathcal{R}_{T_2}]] \\ &= E[f_1(L_{T_1}(Q(T_1))) E[f_2(L_{T_2}(Q(T_2)))] | \mathcal{F}_{T_2}, \mathcal{R}_{T_2}]] \\ &= E[f_1(L_{T_1}(Q(T_1)))] E[f_2(V)] = E[f_1(V)] E[f_2(V)]. \end{aligned}$$

The same argument extends to any subset of arrival times, and thus the iid property follows.

About property (2): Note that in the calculations involving property (1), the actual values of the arrival times T , T_1 and T_2 are irrelevant. The iid property of the service times is established path-by-path conditional on the observed realization \mathcal{T}^0 . Thus, the independence of the arrival process and service times follows immediately.

A.2 Proof of technical lemmas of monotonicity

Proof of Lemma 1. Both facts are standard; the first one can be easily shown using induction. Specifically, we first notice that $W_k(T_k^0; w^+) = w^+ > w^- = W_k(T_n^0; w^-)$. Suppose that $W_k(T_n^0; w^+) \geq W_k(T_n^0; w^-)$ for some $n \geq k$, then

$$\begin{aligned} W_k(T_n^{0,+}; w^+) &= \mathcal{S} \left((W_k(T_n^0; w^+) + V_n \mathbf{e}_1 - A_n \mathbf{1})^+ \right) \\ &\geq \mathcal{S} \left((W_k(T_n^0; w^-) + V_n \mathbf{e}_1 - A_n \mathbf{1})^+ \right) = W_k(T_n^{0,+}; w^-). \end{aligned}$$

For inequality (3.9), we note that $W_k(T_{k'}^0; 0) \geq W_{k'}(T_{k'}^0; 0) = 0$, and therefore, due to (3.8), we have that

$$W_k(T_n^0; 0) = W_{k'}(T_n^0; W_k(T_{k'}^0; 0)) \geq W_{k'}(T_n^0; 0).$$

□

Proof of Lemma 2. This fact follows immediately by induction from Eqs. (3.4) and (3.7) using the fact that $\Xi_n \geq 0$. □

Proof of Lemma 3. We first prove the inequality $Q_u(t - u; z^+) \leq Q_v(t)$. Note that $U^i(u) > 0$ for all u (the forward residual life time process is right continuous), so the initial condition r indicates that all the servers are busy (operating) and the initial $q \geq 0$ customers will leave the queue (i.e., enter service) at the same time as those in the vacation system under the evolution of $Z_u(\cdot; z^+)$. Now, let us write $N = \inf\{n : T_n^0 \geq u\}$ (in words, the next arriving customer at or after u arrives at time T_N^0). It is easy to see that $\mathcal{S}(U(T_N^0)) \geq R_u(T_N^0 - u; z^+)$; to wit, if T_N^0 occurs before any of the servers becomes idle, then we have equality, and if T_N^0 occurs after, say, $l \geq 1$ servers

become idle, then $R_u(T_N^0 - u; z^+)$ will have l zeroes and the bottom $c - l$ entries will coincide with those of $\mathcal{S}(U(T_N^0))$, which has strictly positive entries. So, if w_N is the Kiefer-Wolfowitz vector observed by the customer arriving at T_N^0 (induced by $Q_u(\cdot - u; z^+)$), then we have $W_v(N) \geq w_N$. By monotonicity of the Kiefer-Wolfowitz vector in the initial condition and because of Lemma 2, we have

$$W_v(k) \geq W_N(T_k^0; W_v(N)) \geq W_N(T_k^0; w_N),$$

for all $k \geq N$, and hence, $T_k^0 + D_k^0 \geq T_k^0 + W_N^{(1)}(T_k^0; w_N)$. Therefore, the departure time from the queue (i.e., initiation of service) of the customer arriving at T_k^0 in the vacation system occurs no earlier than the departure time from the queue of the customer arriving at time T_k^0 in the $GI/GI/c$ queue. Consequently, we conclude that the set of customers waiting in the queue in the $GI/GI/c$ system at time t is a subset of the set of customers waiting in the queue in the vacation system at the same time. Similarly, we consider $Q_u(t - u; z^-) \leq Q_u(t - u; z^+)$, which is easier to establish, since, for $k \geq N$ (with the earlier definition of T_N^0 and w_N),

$$W_N(T_k^0; w_N) \geq W_N(T_k^0; 0),$$

So the set of customers waiting in the queue in the lower bound $GI/GI/c$ system at time t is a subset of the set of customers waiting in the queue in the upper bound $GI/GI/c$ system at the same time. \square

Appendix B

Appendix to Chapter 4

B.1 Detailed algorithm steps in Section 4.2.1

To simulate the process $\{(S_n^{(r)}, \bar{W}_{-n}^0) : 0 \leq n \leq N\}$ with the stopping time N defined in (4.17) as

$$N = \inf \left\{ n \geq 0 : \bar{W}_{-n}^0 = \max_{k \geq n} S_k^{(r)} - S_n^{(r)} = \mathbf{0} \right\},$$

we must sample the running time maxima (entry by entry) of the c -dimensional random walk

$$S_n^{(r)} = \sum_{i=1}^n \Delta_{-i} = \sum_{i=1}^n (V_{-i} \mathbf{u}_{-i} - A_{-i} \mathbf{1}) \quad n \geq 0.$$

We will find a sequence of random times $\{N_n : n \geq 1\}$ such that $\max_{n \leq k \leq N_n} S_k^{(r)} \geq \max_{k \geq N_n} S_k^{(r)}$.

Hence, we will be able to find the running time maxima by only sampling the random walk on a finite time interval, i.e., N_n is such that

$$\max_{k \geq n} S_k^{(r)} = \max_{n \leq k \leq N_n} S_k^{(r)}.$$

To achieve this, we first decompose the random walk into two random walks, and then construct a sequence of “milestone” events for each of these two random walks to detect N_n . We will elaborate the detailed implementations in the following context.

Because of the stability condition $\rho = \lambda/(c\mu) < 1$, we can find some value $a \in (1/\mu, c/\lambda)$. For

any $n \geq 0$, define

$$X_{-n} = \sum_{j=1}^n (V_{-j} - a) \mathbf{u}_{-j}, \quad (\text{B.1})$$

$$Y_{-n} = \sum_{j=1}^n (a\mathbf{u}_{-j} - A_{-j}\mathbf{1}), \quad (\text{B.2})$$

hence $S_n^{(r)} = \sum_{j=1}^n \Delta_{-j} = X_{-n} + Y_{-n}$ and $\max_{k \geq n} S_k^{(r)} = \max_{k \geq n} (X_{-n} + Y_{-n})$.

For all $n \geq 0$, let

$$N_n^X = \inf\{n' \geq n : \max_{k \geq n'} X_{-k} \leq X_{-n}\}, \quad (\text{B.3})$$

$$N_n^Y = \inf\{n' \geq n : \max_{k \geq n'} Y_{-k} \leq Y_{-n}\}, \quad (\text{B.4})$$

$$N_n = \max\{N_n^X, N_n^Y\}. \quad (\text{B.5})$$

Then, by the definitions above,

$$\max_{k \geq N_n} S_k^{(r)} \leq \max_{k \geq N_n} X_{-k} + \max_{k \geq N_n} Y_{-k} \leq X_{-n} + Y_{-n} = S_n^{(r)}.$$

Therefore, to get the running-time maximum $\max_{k \geq n} S_k^{(r)}$ for each $n \geq 0$, we only need to sample the random walk from step n to N_n , because

$$\max_{k \geq n} S_k^{(r)} = \max\left\{\max_{n \leq k \leq N_n} S_k^{(r)}, \max_{n \geq N_n} S_k^{(r)}\right\} = \max_{n \leq k \leq N_n} S_k^{(r)}.$$

Next, we describe how to sample N_n along with the multi-dimensional random walks $\{X_{-n} : n \geq 0\}$ and $\{Y_{-n} : n \geq 0\}$.

B.1.1 Simulation algorithm for the process $\{Y_{-n} : n \geq 0\}$

We first consider simulating the c -dimensional random walk $\{Y_{-n} : n \geq 0\}$ with $Y_0 = \mathbf{0}$. For each $j \geq 1$, $E[a\mathbf{u}_{-j} - A_{-j}\mathbf{1}] < \mathbf{0}$, we can simulate the running time maximum $\max_{k \geq n} Y_{-k}$ jointly with the path $\{Y_{-k} : 0 \leq k \leq n\}$ via the exponential change of measure method developed in [Blanchet and Chen, 2015], with the following assumptions.

Assumption 6. *There exists $\theta > \mathbf{0}$, $\theta \in \mathbb{R}^c$ such that*

$$E[\exp(\theta^T(a\mathbf{u}_{-j} - A_{-j}\mathbf{1}))] < \infty.$$

Assumption 7. *Suppose that in every dimension $i = 1, \dots, c$, there exists $\theta^* \in (0, \infty)$ such that*

$$\phi_i(\theta^*) := \log E[\exp(\theta^*(aI(U_{-j} = i) - A_{-j}))] = 0.$$

Because for each $j \geq 1$, $aI(U_{-j} = i) - A_{-j}$ are marginally identically distributed across i , so θ^* would work for all $i = 1, \dots, c$.

Remark 4. *Assumption 7 is known as Cramer's condition in the large deviations literature and it is a strengthening of Assumption 6. We shall explain briefly at the end of this section that it is possible to relax this assumption to Assumption 6 by modifying the algorithm a bit without affecting the exactness/computational effort of the algorithm. For the moment we continue to describe the main algorithmic idea under Assumption 7.*

For any $\mathbf{s} \in \mathbb{R}^c$ and $\mathbf{b} \in \mathbb{R}_+^c$ define

$$T_{\mathbf{b}} = \inf\{n \geq 0 : Y_{-n}(i) > b(i) \text{ for some } i \in \{1, \dots, c\}\}, \quad (\text{B.6})$$

$$T_{-\mathbf{b}} = \inf\{n \geq 0 : Y_{-n}(i) < -b(i) \text{ for all } i = 1, \dots, c\}, \quad (\text{B.7})$$

$$P_{\mathbf{s}}(\cdot) = P(\cdot | Y_0 = \mathbf{s}). \quad (\text{B.8})$$

We will use these definitions in Algorithm LTGM given in Section B.1.1.1.

We next construct a sequence of upward and downward “milestone” events for this multi-dimensional random walk. Let

$$m = \lceil \log(c)/\theta^* \rceil. \quad (\text{B.9})$$

Define $D_0 = 0$ and $\Gamma_0 = \infty$. For $k \geq 1$, let

$$D_k = \inf\{n \geq D_{k-1} \vee \Gamma_{k-1} I(\Gamma_{k-1} < \infty) : Y_{-n}(i) < Y_{-D_{k-1}}(i) - m \text{ for all } i\}, \quad (\text{B.10})$$

$$\Gamma_k = \inf\{n \geq D_k : Y_{-n}(i) > Y_{-D_k}(i) + m \text{ for some } i\}, \quad (\text{B.11})$$

where m is defined in (B.9). Note that by convention, $\Gamma_k I(\Gamma_k < \infty) = 0$ if $\Gamma_k = \infty$ for any $k \geq 0$.

We let $B \in \mathbb{R}^c$, initially set as $(\infty, \dots, \infty)^T \in \mathbb{R}^c$, to be the running time upper bound of process $\{Y_{-n} : n \geq 0\}$. Let $\mathbf{m} = m\mathbf{1}$. From the construction of “milestone” events in (B.10) and (B.11), we know that if $\Gamma_k = \infty$ for some $k \geq 1$, the process will never cross over the level $Y_{-D_k} + \mathbf{m}$ after D_k coordinate-wise, i.e., for $i = 1, \dots, c$,

$$Y_{-n}(i) \leq Y_{-D_k}(i) + m, \quad \forall n \geq D_k.$$

Hence, in this case we update the upper bound vector $B = Y_{-D_k} + \mathbf{m}$.

B.1.1.1 Global maximum simulation

Define

$$\Lambda = \inf\{D_k : \Gamma_k = \infty, k \geq 1\}. \quad (\text{B.12})$$

By the construction of “milestone” events, for all $n \geq \Lambda$

$$Y_{-n} \leq Y_{-\Lambda} + \mathbf{m} < \mathbf{0} = Y_0.$$

Hence, we can evaluate the global maximum level of the process $\{Y_{-n} : n \geq 0\}$ to be

$$M_0 := \max_{k \geq 0} Y_{-k} = \max_{0 \leq k \leq \Lambda} Y_{-k}, \quad (\text{B.13})$$

and we give the detailed sampling procedure in the following algorithm. The algorithm has elements, such as sampling from $P_{\mathbf{0}}(T_{\mathbf{m}} < \infty)$, which will be explained in the sequel.

Algorithm LTGM: Simulate global maximum of c -dimensional process $\{Y_{-n} : n \geq 0\}$ jointly with the sub-path and the subsequence of “milestone” events.

Input: $a \in (1/\mu, c/\lambda)$ satisfies Assumption 7, m as in (B.9).

1. (*Initialization*) Set $n = 0$, $Y_0 = \mathbf{0}$, $\mathbf{D} = [0]$, $\mathbf{\Gamma} = [\infty]$, $L = \mathbf{0}$ and $B = \infty \mathbf{1}$.
2. Generate $U \sim \text{Unif}\{1, \dots, c\}$ and let $\mathbf{u} = (I(U = 1), \dots, I(U = c))^T$. Independently sample $A \sim G$. Set $n = n + 1$, $Y_{-n} = Y_{-(n-1)} + a\mathbf{u} - A\mathbf{1}$, $U_{-n} = U$ and $A_{-n} = A$.
3. If there is some $1 \leq i \leq c$ such that $Y_{-n}(i) \geq L(i) - m$, then go to Step 2; otherwise set $\mathbf{D} = [\mathbf{D}, n]$ and $L = Y_{-n}$.
4. Independently sample $J \sim \text{Ber}(P_{\mathbf{0}}(T_{\mathbf{m}} < \infty))$.
5. If $J = 1$, simulate a new conditional path $\{(\tilde{Y}_{-k}, \tilde{U}_{-k}, \tilde{A}_{-k}) : 1 \leq k \leq T_{\mathbf{m}}\}$ with $\tilde{Y}_0 = \mathbf{0}$, following the conditional distribution of $\{Y_{-k} : 0 \leq k \leq T_{\mathbf{m}}\}$ given $T_{\mathbf{m}} < \infty$. Set $Y_{-(n+k)} = Y_{-n} + \tilde{Y}_{-k}$, $U_{-(n+k)} = \tilde{U}_{-k}$, $A_{-(n+k)} = \tilde{A}_{-k}$ for $1 \leq k \leq T_{\mathbf{m}}$. Set $n = n + T_{\mathbf{m}}$, $\mathbf{\Gamma} = [\mathbf{\Gamma}, n]$. Go to Step 2.
6. If $J = 0$, set $\Lambda = n$, $\mathbf{\Gamma} = [\mathbf{\Gamma}, \infty]$ and $B = L + \mathbf{m}$.

7. Output $\{(Y_{-k}, U_{-k}, T_{-k}) : 1 \leq k \leq \Lambda\}$, \mathbf{D} , $\mathbf{\Gamma}$ and global maximum $M_0 = \max_{0 \leq k \leq \Lambda} Y_{-k}$.

Now we explain how to execute Steps 4 and 5 in the previous algorithm. The procedure is similar to the multi-dimensional procedure given in [Blanchet and Chen, 2015], so we describe it briefly here. As $P_0(\cdot)$ denotes the canonical probability, we let $P_0^*(\cdot) = P_0(\cdot | T_{\mathbf{m}} < \infty)$. Our goal is to simulate from the conditional law of $\{Y_{-k} : 0 \leq k \leq T_{\mathbf{m}}\}$ given that $T_{\mathbf{m}} < \infty$ and $Y_0 = \mathbf{0}$, i.e., to simulate from P_0^* . We will use acceptance/rejection by letting $P'_0(\cdot)$ denote the proposal distribution. A typical element ω' sampled under $P'_0(\cdot)$ is of the form $\omega' = ((Y_{-k} : k \geq 0), index)$, where $index \in \{1, \dots, c\}$ and it indicates the direction we pick to do exponential tilting. Given the value of $index$, the process $(Y_{-k} : k \geq 0)$ remains a random walk. We now describe P'_0 by explaining how to sample ω' . First,

$$P'_0(index = i) := \frac{1}{c}. \quad (\text{B.14})$$

Then, conditioning on $index = i$, for every set $\Omega \in \sigma(\{Y_{-k} : 0 \leq k \leq n\})$,

$$P'_0(\Omega | index = i) = E_{\mathbf{0}}(\exp(\theta^* Y_{-n}(i)) I_{\Omega}). \quad (\text{B.15})$$

To obtain the induced distribution for U and A , we study the moment generating function induced by definition (B.15). Given $\eta \in \mathbb{R}^c$ in a neighborhood of the origin,

$$\frac{E_{\mathbf{0}} \exp(\eta^T (a\mathbf{u} - A\mathbf{1}) + \theta^* e_i^T (a\mathbf{u} - A\mathbf{1}))}{E_{\mathbf{0}} \exp(\theta^* e_i^T (a\mathbf{u} - A\mathbf{1}))} = \frac{E_{\mathbf{0}} \exp((\eta + \theta^* e_i)^T a\mathbf{u})}{E_{\mathbf{0}} \exp(\theta^* e_i^T a\mathbf{u})} \cdot \frac{E_{\mathbf{0}} \exp(-(\eta + \theta^* e_i)^T A\mathbf{1})}{E_{\mathbf{0}} \exp(-\theta^* e_i^T A\mathbf{1})}.$$

The previous expression indicates that under $P'_0(\cdot)$, A and U are independent. Moreover, we have

$$E_{\mathbf{0}}[\exp(\theta^* e_i^T a\mathbf{u})] = \frac{\exp(\theta^* a) + c - 1}{c}.$$

Therefore,

$$P'_0(U = j | index = i) = \begin{cases} \frac{\exp(\theta^* a)}{\exp(\theta^* a) + c - 1} & \text{if } j = i \\ \frac{1}{\exp(\theta^* a) + c - 1} & \text{if } j \neq i \end{cases}. \quad (\text{B.16})$$

On the other hand, conditional on $index = i$, the distribution of a generic interarrival time A is obtained by exponential tilting such that

$$\begin{aligned} dP_0(A | index = i) &= dP_0(A) \cdot \frac{\exp(-\theta^* A)}{E_{\mathbf{0}} \exp(-\theta^* A)} \\ &= dP_0(A) \cdot \frac{\exp(a\theta^*) + c - 1}{c \exp(\theta^* A)}, \end{aligned} \quad (\text{B.17})$$

where the second equation follows from Assumption 7.

Following Assumption 7, and because $\text{Var}(aI(U_{-j} = i) - A_{-j}) > 0$, by convexity,

$$\begin{aligned} E'_{\mathbf{0}}(Y_{-n}(\text{index})) &= \sum_{i=1}^c E_{\mathbf{0}}(Y_{-n}(i) \exp(\theta^* Y_{-n}(i))) P'_{\mathbf{0}}(\text{index} = i) \\ &= \frac{1}{c} \sum_{i=1}^c \frac{d\phi_i(\theta^*)}{d\theta} > 0, \end{aligned}$$

so $Y_{-n}(\text{index}) \rightarrow \infty$ as $n \rightarrow \infty$ almost surely under $P'_{\mathbf{0}}(\cdot)$, hence $T_{\mathbf{m}} < \infty$ with probability one under $P'_{\mathbf{0}}(\cdot)$. Now, to verify that $P_{\mathbf{0}}(\cdot)$ is a valid proposal for acceptance/rejection method, we must verify that $dP_{\mathbf{0}}^*/dP'_{\mathbf{0}}$ is bounded by a constant, i.e.,

$$\begin{aligned} &\frac{dP_{\mathbf{0}}^*}{dP'_{\mathbf{0}}}(Y_{-k} : 0 \leq k \leq T_{\mathbf{m}}) \\ &= \frac{1}{P_0(T_{\mathbf{m}} < \infty)} \times \frac{dP_0}{dP'_{\mathbf{0}}}(Y_{-k} : 0 \leq k \leq T_{\mathbf{m}}) \\ &= \frac{1}{P_0(T_{\mathbf{m}} < \infty)} \times \frac{1}{\sum_{i=1}^c w_i \exp(\theta^* Y_{-T_{\mathbf{m}}}(i))} \\ &\leq \frac{1}{P_0(T_{\mathbf{m}} < \infty)} \times \frac{c}{\exp(\theta^* m)} \\ &< \frac{1}{P_0(T_{\mathbf{m}} < \infty)}, \end{aligned}$$

where the last inequality is guaranteed by (B.9). So, acceptance/rejection is valid.

Moreover, the overall probability of accepting the proposal is precisely $P_{\mathbf{0}}(T_{\mathbf{m}} < \infty)$. Thus, we not only execute Step 5, but simultaneously also Step 4. We use this acceptance/rejection method to replace Steps 4 and 5 in Algorithm LTGM as follows:

- 4' Sample $\left\{ \left(\tilde{Y}_{-k}, \tilde{U}_{-k}, \tilde{A}_{-k} \right) : 0 \leq k \leq T_{\mathbf{m}} \right\}$ with $\tilde{Y}_0 = \mathbf{0}$ from $P'_{\mathbf{0}}(\cdot)$ as indicated via (B.14), (B.16) and (B.17). Sample a Bernoulli J with success probability

$$\frac{c}{\sum_{i=1}^c \exp(\theta^* \tilde{Y}_{-T_{\mathbf{m}}}(i))}.$$

- 5' If $J = 1$, set $Y_{-(n+k)} = Y_{-n} + \tilde{Y}_{-k}$, $U_{-(n+k)} = \tilde{U}_{-k}$, $A_{-(n+k)} = \tilde{A}_{-k}$ for $1 \leq k \leq T_{\mathbf{m}}$. Set $n = n + T_{\mathbf{m}}$ and $\mathbf{\Gamma} = [\mathbf{\Gamma}, n]$. Go to Step 2.

B.1.1.2 Simulate $\{Y_{-n} : n \geq 0\}$ with “milestone” events

In this section we provide an algorithm to sequentially simulate the multi-dimensional random walk $\{Y_{-n} : n \geq 0\}$ along with its downward and upward “milestone” events as defined in (B.10) and (B.11). We first extend Lemma 3 in [Blanchet and Sigman, 2011] to multi-dimensional version as follows.

Lemma 9. *Let $0 < \mathbf{a} < \mathbf{b} \leq \infty \mathbf{1}$ (coordinate-wise) and consider any sequence of bounded positive measurable functions $f_k : \mathbb{R}_{c \times (k+1)} \rightarrow [0, \infty)$,*

$$\begin{aligned} & E_0(f_{T_{-\mathbf{a}}}(Y_0, \dots, Y_{-T_{-\mathbf{a}}}) | T_{\mathbf{b}} = \infty) \\ &= \frac{E_0(f_{T_{-\mathbf{a}}}(Y_0, \dots, Y_{-T_{-\mathbf{a}}}) \cdot I(Y_{-j}(i) \leq b(i), 0 \leq j < T_{-\mathbf{a}}, 1 \leq i \leq c)) \cdot P_{Y_{-T_{-\mathbf{a}}}}(T_{\mathbf{b}} = \infty)}{P_0(T_{\mathbf{b}} = \infty)}. \end{aligned}$$

Therefore, if $P_0^{**}(\cdot) := P_0(\cdot | T_{\mathbf{b}} = \infty)$, then

$$\frac{dP_0^{**}}{dP_0} = \frac{I(Y_{-j}(i) \leq b(i), \forall j < T_{-\mathbf{a}}, 1 \leq i \leq c) \cdot P_{Y_{-T_{-\mathbf{a}}}}(T_{\mathbf{b}} = \infty)}{P_0(T_{\mathbf{b}} = \infty)} \leq \frac{1}{P_0(T_{\mathbf{b}} = \infty)}. \quad (\text{B.18})$$

Lemma 9 enables us to sample a downward patch by using the acceptance/rejection method with the nominal distribution P_0 as proposal. Suppose our current position is Y_{-D_j} (for some $j \geq 1$) and we know that the process will never go above the upper bound B (coordinate-wise). Next we simulate the path up to time D_{j+1} . If we can propose a downward patch $(\tilde{Y}_{-1}, \dots, \tilde{Y}_{-T_{-\mathbf{m}}}) := (Y_{-1}, \dots, Y_{-T_{-\mathbf{m}}})$, under the unconditional probability given $\tilde{Y}_0 = \mathbf{0}$ and $\tilde{Y}_{-k} \leq \mathbf{m}$ for $1 \leq k \leq T_{-\mathbf{m}}$, then we accept it with probability $P_0(T_\sigma = \infty)$, where $\sigma = B - Y_{-D_j} - \tilde{Y}_{-T_{-\mathbf{m}}}$. A more efficient way to sample is to sequentially generate $(\tilde{Y}_{-1}, \dots, \tilde{Y}_{-\Lambda})$ with $\tilde{Y}_0 = \mathbf{0}$ as long as $\mathbf{m}_0 := \max_{0 \leq k \leq \Lambda} \tilde{Y}_{-k} \leq \mathbf{m}$ coordinate-wise, then concatenate the sequence to previously sampled subpath. We give the efficient implementation procedure in the next algorithm.

Algorithm LTRW: Continue to sample the process $\{(Y_{-k}, U_{-k}, A_{-k}) : 0 \leq k \leq n\}$ jointly with the partially sampled “milestone” event lists \mathbf{D} and $\mathbf{\Gamma}$, until some stopping criteria are met.

Input: a, m , previously sampled partial process $\{(Y_{-j}, U_{-j}, A_{-j}) : 0 \leq j \leq l\}$, partial “milestone” sequences \mathbf{D} and $\mathbf{\Gamma}$, and the stopping criteria \mathcal{H} .

(Note that if there is no previous simulated random walk, we initialize $l = 0$, $\mathbf{D} = [0]$ and $\mathbf{\Gamma} = [\infty]$.)

1. Set $n = l$. If $n = 0$, call Algorithm LTGM to get Λ , $\{(Y_{-k}, U_{-k}, A_{-k}) : 0 \leq k \leq \Lambda\}$, \mathbf{D} and $\mathbf{\Gamma}$. Set $n = \Lambda$.
2. While the stopping criteria \mathcal{H} are not satisfied,
 - (a) Call Algorithm LTGM to get $\tilde{\Lambda}$, $\{(\tilde{Y}_{-j}, \tilde{U}_{-j}, \tilde{A}_{-j}) : 0 \leq j \leq \tilde{\Lambda}\}$, $\tilde{\mathbf{D}}$, $\tilde{\mathbf{\Gamma}}$ and \tilde{M}_0 .
 - (b) If $\tilde{M}_0 \leq \mathbf{m}$, accept the proposed sequence and concatenate it to the previous sub-path, i.e., set $Y_{-(n+j)} = Y_{-n} + \tilde{Y}_{-j}$, $U_{-(n+j)} = \tilde{U}_{-j}$, $A_{-(n+j)} = \tilde{A}_{-j}$ for $1 \leq j \leq \tilde{\Lambda}$. Update the sequences of “milestone” events to be $\mathbf{D} = [\mathbf{D}, n + \tilde{\mathbf{D}}(2 : \text{end})]$, $\mathbf{\Gamma} = [\mathbf{\Gamma}, n + \tilde{\mathbf{\Gamma}}(2 : \text{end})]$ and set $n = n + \tilde{\Lambda}$.
3. Output $\{(Y_{-k}, U_{-k}, A_{-k}) : 0 \leq k \leq n\}$ with updated “milestone” event sequences \mathbf{D} and $\mathbf{\Gamma}$.

For $n \geq 0$, define

$$d_1(n) = \inf\{D_k \geq n : Y_{-D_k} \leq Y_{-n}\}, \quad (\text{B.19})$$

$$d_2(n) = \inf\{D_k > d_1(n) : \Gamma_k = \infty\}, \quad (\text{B.20})$$

and $d_2(n)$ is an upper bound of N_n^Y defined in (B.4) because

$$\max_{k \geq d_2(n)} Y_{-k} \leq Y_{-d_2(n)} + \mathbf{m} < Y_{-d_1(n)} \leq Y_{-n}.$$

Remark 5. *Although Assumption 7 is a strengthening of Assumption 6, we can accommodate our algorithms under Assumption 6. The implementation details are the same as that mentioned in the remark section on page 15 of [Blanchet and Chen, 2015].*

B.1.2 Simulation algorithm for the process $\{X_{-n} : n \geq 0\}$

Recall from (B.1) that for $n \geq 0$,

$$X_{-n}(i) = \sum_{j=1}^n (V_{-j} - a) I(U_{-j} = i) \quad \text{for } i = 1, \dots, c.$$

Define

$$N_k(i) = \sum_{j=1}^k I(U_{-j} = i), \quad (\text{B.21})$$

$$L_n(i) = \inf\{k \geq 0 : N_k(i) = n\} \quad (L_0(i) = 0), \quad (\text{B.22})$$

$$\hat{V}_{-n}^{(i)} = V_{-L_n(i)}, \quad (\text{B.23})$$

for $k \geq 0$, $n \geq 0$ and $i = 1, \dots, c$. $N_k(i)$ denotes the total number of customers routed to server i among the first k arrivals counting backwards in time. $L_n(i)$ denotes the index of the n -th customer that gets routed to server i in the common arrival stream, counting backwards in time. $\hat{V}_{-n}^{(i)}$ denotes the service time of the n -th customer that gets routed to server i , counting backwards in time.

For each $i = 1, \dots, c$, let $\{\hat{X}_{-n}^{(i)} : n \geq 0\}$ with $\hat{X}_0^{(i)} = 0$ be an auxiliary process such that

$$\hat{X}_{-n}^{(i)} := \sum_{j=1}^n \left(\hat{V}_{-j}^{(i)} - a \right) = X_{-L_n(i)}^{(i)}. \quad (\text{B.24})$$

For $n \geq 0$ and $1 \leq i \leq c$, define

$$\hat{N}_n(i) = \inf \left\{ n' \geq N_n(i) : \max_{k \geq n'} \hat{X}_{-k}^{(i)} \leq \hat{X}_{-N_n(i)}^{(i)} \right\}, \quad (\text{B.25})$$

hence by definition, in (B.3), we have

$$N_n^X = \max \left\{ L_{\hat{N}_n(1)}(1), \dots, L_{\hat{N}_n(c)}(c) \right\}. \quad (\text{B.26})$$

First we develop simulation algorithms for each of the c one-dimensional auxiliary processes $\{\hat{X}_{-n}^{(i)} : n \geq 0\} : 1 \leq i \leq c\}$. Next we use the common server allocation sequence $\{U_{-n} : n \geq 0\}$ (sampled jointly with the process $\{Y_{-n} : n \geq 0\}$ in Section B.1.1) with (B.21), (B.22) and (B.23) to find N_n^X via (B.26) for each $n \geq 0$.

“Milestone” construction and global maximum simulation For each one-dimensional auxiliary process $\{\hat{X}_{-n}^{(i)} : n \geq 0\}$ with $i = 1, \dots, c$, we adopt the algorithm developed in [Blanchet and Wallwater, 2015] by choosing any $m' > 0$ and $L' \geq 1$ properly and define the sequences of upward and downward “milestone” events by letting $D_0^{(i)} = 0$, $\Gamma_0^{(i)} = \infty$, and for $j \geq 1$,

$$D_j^{(i)} = \inf \{ n^{(i)} \geq \Gamma_{j-1}^{(i)} I(\Gamma_{j-1}^{(i)} < \infty) \vee D_{j-1}^{(i)} : \hat{X}_{-n^{(i)}}^{(i)} < \hat{X}_{-D_{j-1}^{(i)}}^{(i)} - L' m' \}, \quad (\text{B.27})$$

$$\Gamma_j^{(i)} = \inf \{ n^{(i)} \geq D_j^{(i)} : \hat{X}_{-n^{(i)}}^{(i)} - \hat{X}_{-D_j^{(i)}}^{(i)} > m' \}, \quad (\text{B.28})$$

with the convention that if $\Gamma_j^{(i)} = \infty$, then $\Gamma_j^{(i)} I(\Gamma_j^{(i)} < \infty) = 0$ for any $j \geq 0$.

For each $i = 1, \dots, c$, define

$$\Lambda^{(i)} = \inf \{ D_k^{(i)} : \Gamma_k^{(i)} = \infty, k \geq 1 \}. \quad (\text{B.29})$$

By the “milestone” construction in (B.27) and (B.28), for all $n \geq \Lambda^{(i)}$,

$$\hat{X}_{-n}^{(i)} \leq \hat{X}_{-\Lambda^{(i)}}^{(i)} + m' < 0 = \hat{X}_0^{(i)}.$$

Therefore we can evaluate the global maximum of the infinite-horizon process $\{\hat{X}_{-n}^{(i)} : n \geq 0\}$ in finite steps, i.e.,

$$M_0^{(i)} := \max_{k \geq 0} \hat{X}_{-k}^{(i)} = \max_{0 \leq k \leq \Lambda^{(i)}} \hat{X}_{-k}^{(i)}. \quad (\text{B.30})$$

We summarize the simulation details in the following algorithm.

Algorithm GGM: Simulate global maximum of the one-dimensional process $\{(\hat{X}_{-n}^{(i)}, \hat{V}_{-n}^{(i)}) : n \geq 0\}$ jointly with the sub-path and the subsequence of “milestone” events.

Input: a, m', L' .

1. (*Initialization*) Set $n = 0$, $\hat{X}_0^{(i)} = 0$, $\mathbf{D}^{(i)} = [0]$, $\mathbf{\Gamma}^{(i)} = [\infty]$, $L^{(i)} = 0$.
2. Generate $V \sim F$. Set $n = n + 1$, $\hat{X}_{-n}^{(i)} = \hat{X}_{-(n-1)}^{(i)} + V$ and $\hat{V}_{-n}^{(i)} = V$.
3. If $\hat{X}_{-n}^{(i)} \geq L^{(i)} - L'm'$, go to Step 2; otherwise set $\mathbf{D}^{(i)} = [\mathbf{D}^{(i)}, n]$ and $L^{(i)} = \hat{X}_{-n}^{(i)}$.
4. Call Algorithm 1 on page 10 of [Blanchet and Wallwater, 2015] and obtain (J, ω) .
5. If $J = 1$, set $\hat{X}_{-(n+l)}^{(i)} = L^{(i)} + \omega(l)$, $\hat{S}_{-(n+l)}^{(i)} = \hat{X}_{-(n+l)}^{(i)} - \hat{X}_{-(n+l-1)}^{(i)} + a$ for $l = 1, \dots, \text{length}(\omega)$. Set $n = n + \text{length}(\omega)$, $\mathbf{\Gamma}^{(i)} = [\mathbf{\Gamma}^{(i)}, n]$ and go to Step 2.
6. If $J = 0$, set $\Lambda^{(i)} = n$, $\mathbf{\Gamma}^{(i)} = [\mathbf{\Gamma}^{(i)}, \infty]$.
7. Output $\{(\hat{X}_{-k}^{(i)}, \hat{V}_{-k}^{(i)}) : 1 \leq k \leq \Lambda^{(i)}\}$, $\mathbf{D}^{(i)}$, $\mathbf{\Gamma}^{(i)}$ and global maximum $M_0^{(i)} = \max_{0 \leq k \leq \Lambda^{(i)}} \hat{X}_{-k}^{(i)}$.

B.1.2.1 Simulate $\{X_{-n} : n \geq 0\}$ with “milestone” events

In this section, we first explain how to sample the auxiliary one-dimensional processes $\{\hat{X}_{-n}^{(i)} : n \geq 0\}$ along with the “milestone” events defined in (B.27) and (B.28). Next we will need the service allocation information $\{U_{-n} : n \geq 0\}$, from the simulation procedure of process $\{Y_{-n} : n \geq 0\}$, to recover the multi-dimensional process of interest $\{X_{-n} : n \geq 0\}$ via Eq. (B.24).

The following algorithm gives the the sampling procedure for each auxiliary one-dimensional process $\{\hat{X}_{-n}^{(i)} : n \geq 0\}$ for $i = 1, \dots, c$. The simulation steps are the same as the procedure given in Algorithm 3 on page 16 of [Blanchet and Wallwater, 2015].

Algorithm GRW: Continue to sample the process $\{(\hat{X}_{-k}^{(i)}, \hat{V}_{-k}^{(i)}) : 0 \leq k \leq n\}$ jointly with the partially sampled “milestone” event lists $\mathbf{D}^{(i)}$ and $\mathbf{\Gamma}^{(i)}$, until a stopping criteria is met.

Input: a, m', L' , previously sampled partial process $\{(\hat{X}_{-j}^{(i)}, \hat{S}_{-j}^{(i)}) : 0 \leq j \leq l\}$, partial “milestone” sequences $\mathbf{D}^{(i)}$ and $\mathbf{\Gamma}^{(i)}$, and stopping criteria $\mathcal{H}^{(i)}$.

(Note that if there is no previously simulated random walk, we initialize $l = 0$, $\mathbf{D}^{(i)} = [0]$ and $\mathbf{\Gamma}^{(i)} = [\infty]$.)

1. Set $n = l$. If $n = 0$, call Algorithm GGM to get $\Lambda^{(i)}$, $\{(\hat{X}_{-k}^{(i)}, \hat{V}_{-k}^{(i)}) : 0 \leq k \leq \Lambda^{(i)}\}$, $\mathbf{D}^{(i)}$ and $\mathbf{\Gamma}^{(i)}$. Set $n = \Lambda^{(i)}$.
2. While the stopping criteria $\mathcal{H}^{(i)}$ are not satisfied,
 - (a) Call Algorithm GGM to get $\tilde{\Lambda}^{(i)}$, $\{(\tilde{X}_{-j}^{(i)}, \tilde{V}_{-j}^{(i)}) : 0 \leq j \leq \tilde{\Lambda}^{(i)}\}$, $\tilde{\mathbf{D}}$, $\tilde{\mathbf{\Gamma}}$ and $\tilde{M}_0^{(i)}$.
 - (b) If $\tilde{M}_0^{(i)} \leq m'$, accept the proposed sequence and concatenate it to the previous sub-path, i.e., set $\hat{X}_{-(n+j)}^{(i)} = \hat{X}_{-n}^{(i)} + \tilde{X}_{-j}^{(i)}$, $\hat{V}_{-(n+j)}^{(i)} = \tilde{V}_{-j}^{(i)}$ for $1 \leq j \leq \tilde{\Lambda}^{(i)}$. Update the sequences of “milestone” events to be $\mathbf{D}^{(i)} = [\mathbf{D}^{(i)}, n + \tilde{\mathbf{D}}^{(i)}(2 : \text{end})]$, $\mathbf{\Gamma}^{(i)} = [\mathbf{\Gamma}^{(i)}, n + \tilde{\mathbf{\Gamma}}^{(i)}(2 : \text{end})]$ and set $n = n + \tilde{\Lambda}^{(i)}$.
3. Output $\{(\hat{X}_{-k}^{(i)}, \hat{V}_{-k}^{(i)}) : 0 \leq k \leq n\}$ with updated “milestone” event sequences $\mathbf{D}^{(i)}$ and $\mathbf{\Gamma}^{(i)}$.

With the service allocation information $\{U_{-n} : n \geq 0\}$, we can construct the c -dimensional process $\{X_{-n} : n \geq 0\}$ ($X_0 = \mathbf{0}$) from the auxiliary processes $\{(\hat{X}_{-n}^{(i)}, \hat{V}_{-n}^{(i)}) : n \geq 0\}$, $i = 1, \dots, c$. For $n \geq 1$,

$$V_{-n} = \hat{V}_{\sum_{j=1}^n I(U_{-j}=U_{-n})}^{(U_{-n})}, \quad (\text{B.31})$$

$$X_{-n}(i) = \begin{cases} X_{-(n-1)}(i) & \text{if } i \neq U_{-n} \\ X_{-(n-1)}(i) + V_{-n} - a & \text{if } i = U_{-n} \end{cases}. \quad (\text{B.32})$$

By the definition of “milestone” events in (B.27) and (B.28), for each $n \geq 0$, let

$$d_1^{(i)}(n) = \inf\{D_k^{(i)} \geq n : \hat{X}_{-D_k^{(i)}}^{(i)} \leq \hat{X}_{-n}^{(i)}\}, \quad (\text{B.33})$$

$$d_2^{(i)}(n) = \inf\{D_k^{(i)} > d_1^{(i)}(n) : \Gamma_k^{(i)} = \infty\}. \quad (\text{B.34})$$

Since

$$\max_{k \geq d_2^{(i)}(N_n(i))} \hat{X}_{-k}^{(i)} \leq \hat{X}_{-d_2^{(i)}(N_n(i))}^{(i)} + m' < \hat{X}_{-d_1^{(i)}(N_n(i))}^{(i)} \leq \hat{X}_{-N_n(i)}^{(i)},$$

we conclude that $\hat{N}_n(i) \leq d_2^{(i)}(N_n(i))$ and hence

$$N_n^X \leq \max\{L_{d_2^{(1)}(N_n(1))}(1), \dots, L_{d_2^{(c)}(N_n(c))}(c)\}.$$

B.1.3 Simulation algorithm for $\{S_n^{(r)} : n \geq 0\}$ and coalescence detection

We shall combine the simulation algorithms in Section B.1.1 and Section B.1.2 for processes $\{((\hat{X}_{-n}^{(i)}, \hat{V}_{-n}^{(i)}) : n \geq 0), 1 \leq i \leq c\}$ and $\{(Y_{-n}, U_{-n}, A_{-n}) : n \geq 0\}$ together to exactly simulate the multi-dimensional random walk $\{S_n^{(r)} : n \geq 0\}$ until coalescence time N defined in (4.17). To detect the coalescence, we start from $n = 0$ to compute $d_2(n)$ and $d_2^{(i)}(N_n(i))$ (as defined in (B.20) and (B.34) respectively). If

$$\max_{n \leq k \leq d_2(n)} Y_{-k} = Y_{-n}, \tag{B.35}$$

and

$$\max_{N_n(i) \leq k \leq d_2^{(i)}(N_n(i))} \hat{X}_{-k}^{(i)} = \hat{X}_{-N_n(i)}^{(i)} \tag{B.36}$$

for all $i = 1, \dots, c$, we set the coalescence time $N \leftarrow n$ and stop. Otherwise we increase n by 1 and repeat the above procedure until the first time that (B.35) and (B.36) are satisfied.

In the following algorithm we give the simulation procedure to detect coalescence while sampling the time-reversed multi-dimensional process $\{S_n^{(r)} : n \geq 0\}$.

Algorithm CD: Sample the coalescence time N jointly with the process $\{S_n^{(r)} : n \geq 0\}$.

Input: a, m, m', L' .

1. (*Initialization*) Set $n = 0$. Set $l = 0, Y_0 = \mathbf{0}, \mathbf{D} = [0], \mathbf{\Gamma} = [\infty]$. Set $l_i = 0, \hat{X}_0^{(i)} = 0, \mathbf{D}^{(i)} = [0], \mathbf{\Gamma}^{(i)} = [\infty]$ for all $i = 1, \dots, c$.
2. Call Algorithm LTRW to further sample $\{(Y_{-j}, U_{-j}, A_{-j}) : 0 \leq j \leq l\}$, \mathbf{D} and $\mathbf{\Gamma}$ with the stopping criteria \mathcal{H} being $\sum_{j=1}^l I(U_{-j} = i) > l_i$ for all $i = 1, \dots, c$ and $Y_{-\mathbf{D}(\text{end}-1)} \leq Y_{-n}$.
3. For each $i = 1, \dots, c$,

- (a) Set $n_i = \sum_{j=1}^n I(U_{-j} = i)$.
 - (b) Call Algorithm GRW to further sample $\{(\hat{X}_{-k}^{(i)}, \hat{V}_{-k}^{(i)}) : 0 \leq k \leq l_i\}$, $\mathbf{D}^{(i)}$ and $\mathbf{\Gamma}^{(i)}$ with the stopping criteria $\mathcal{H}^{(i)}$ being $\sum_{j=1}^l I(U_{-j} = i) \leq l_i$ and $\hat{X}_{-\mathbf{D}^{(i)}(\text{end}-1)}^{(i)} \leq \hat{X}_{-n_i}^{(i)}$.
4. If $\max_{n \leq k \leq \mathbf{D}(\text{end})} Y_{-k} \leq Y_{-n}$ and $\max_{n_i \leq k \leq \mathbf{D}^{(i)}(\text{end})} \hat{X}_{-k}^{(i)} \leq \hat{X}_{-n_i}^{(i)}$ for all $i = 1, \dots, c$, go to next step. Otherwise set $n = n + 1$ and go to Step 2.
 5. For $1 \leq k \leq n$, recover V_{-k} and X_{-k} from the auxiliary processes via Eqs. (B.31) and (B.32).
 6. Output coalescence time $N = n$, the sequence $\{(U_{-k}, A_{-k}, V_{-k}) : 0 \leq k \leq n\}$ and process $\{S_k^{(r)} : 0 \leq k \leq n\}$.

B.2 Proof of propositions

Proof of Proposition 4. Firstly, $E[N] < \infty$ holds true under assumptions $\rho < 1$ and $P(A > V) > 0$ (proved in [Sigman, 1988]). Next we shall prove the computational effort τ has finite expectation as well.

For $n \geq 0$, we have N_n^X , N_n^Y and N_n defined in Eqs. (B.3 - B.5) such that

$$\max_{k \geq N_n} S_k^{(r)} \leq \max_{k \geq N_n} X_{-k} + \max_{k \geq N_n} Y_{-k} \leq X_n + Y_n = S_n^{(r)}.$$

Therefore, in order to evaluate the running-time maximum over the infinite horizon $\max_{k \geq n} S_k^{(r)}$, it only requires sampling from n to N_n backwards in time, i.e.,

$$\max_{k \geq n} S_k^{(r)} = \max\left\{ \max_{n \leq k \leq N_n} S_k^{(r)}, \max_{k \geq N_n} S_k^{(r)} \right\} = \max_{n \geq k \leq N_n} S_k^{(r)}.$$

An easy upper bound for τ is given by $\tilde{\tau} = \sum_{n=0}^N N_n$. By Wald's identity, it suffices to show that $E[N_n] < \infty$ for any $n \geq 0$.

By the ‘‘milestone’’ events construction for multi-dimensional process $\{Y_{-n} : n \geq 0\}$ in (B.10), (B.11) and because $d_2(n)$ is an upper bound of N_n^Y , $E[N_n^Y] \leq E[d_2(n)] < \infty$ follows directly from elementary properties of compound geometric random variables (see Theorem 1 of [Blanchet and Chen, 2015]).

For the other process $\{X_{-n} : n \geq 0\}$, we simulate each of its c entries separately, i.e., $\{\{\hat{X}_{-n}^{(i)} : n \geq 0\} : 1 \leq i \leq c\}$ in Section B.1.2. Eq. (B.26) gives

$$N_n^X = \max\{L_{\hat{N}_n(1)}(1), \dots, L_{\hat{N}_n(c)}(c)\} \leq \sum_{i=1}^c L_{\hat{N}_n(i)}(i),$$

where $\hat{N}_n(i)$ is defined in (B.25). By Theorem 2.2 of [Blanchet and Wallwater, 2015], $E[\hat{N}_n(i)] < \infty$. Because

$$L_{\hat{N}_n(i)}(i) = \inf\{k \geq 0 : \sum_{j=1}^k I(U_{-j} = i) = \hat{N}_n(i)\} \sim \text{NegBinomial}\left(\hat{N}_n(i); 1 - \frac{1}{c}\right) + \hat{N}_n(i),$$

hence

$$E[L_{\hat{N}_n(i)}(i)] = (c-1)E[\hat{N}_n(i)] + E[\hat{N}_n(i)] = cE[\hat{N}_n(i)] < \infty,$$

and

$$E[N_n^X] \leq \sum_{i=1}^c E[L_{\hat{N}_n(i)}(i)] < \infty.$$

Therefore

$$E[N_n] \leq E[N_n^X] + E[N_n^Y] < \infty.$$

□

Proof of Proposition 6. By Wald's identity, it suffices to show that $E[\kappa_+^*] < \infty$ because $E[A] < \infty$. Next we only provide a proof outline here since it follows the same argument as in the proof of Proposition 3.

Firstly, we construct a sequence of events $\{\Omega_k : k \geq 1\}$ which leads to the occurrence of κ_+^* . Secondly, we split the process $\{W_0^u(t_n) : n \geq 0\}$ into cycles with bounded expected cycle length. We also ensure the probability that the event happens during each cycle is bounded from below by a constant, which allows us to bound the number of cycles we need to check before finding κ_+^* by a geometric random variable. Finally we could establish an upper bound for $E[\kappa_+^*]$ by applying Wald's identity again. □