Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

COMPUTATION OF RISK MEASURES IN FINANCE AND PARALLEL REAL-TIME SCHEDULING

by
Yajuan Li

Many application areas employ various risk measures, such as a quantile, to assess risks. For example, in finance, risk managers employ a quantile to help determine appropriate levels of capital needed to be able to absorb (with high probability) large unexpected losses in credit portfolios comprising loans, bonds, and other financial instruments subject to default. This dissertation discusses the computation of risk measures in finance and parallel real-time scheduling.

Firstly, two estimation approaches are compared for one risk measure, a quantile, via randomized quasi-Monte Carlo (RQMC) in an asymptotic setting where the number of randomizations for RQMC grows large, but the size of the low-discrepancy point set remains fixed. In the first method, for each randomization, it computes an estimator of the cumulative distribution function (CDF), which is inverted to obtain a quantile estimator, and the overall quantile estimator is the sample average of the quantile estimators across randomizations. The second approach instead computes a single quantile estimator by inverting one CDF estimator across all randomizations. Because quantile estimators are generally biased, the first method leads to an estimator that does not converge to the true quantile as the number of randomizations goes to infinity. In contrast, the second estimator does, and a central limit theorem is established for it. To get an improvement, we use conditional Monte Carlo (CMC) to obtain a smoother estimate of the distribution function, and we combine this with the second RQMC to further reduce the variance. The result is a much more accurate quantile estimator, whose mean square error can converge even faster than the canonical rate of $O(1/n)$.

Secondly, another risk measure is estimated, namely economic capital (EC), which is defined as the difference between a quantile and the mean of the loss distribution, given a stochastic model for a portfolio's loss over a given time horizon. This work applies measure-specific importance sampling to separately estimate the two components of the EC, which can lead to a much smaller variance than when estimating both terms simultaneously.

Finally, for parallel real-time tasks, the federated scheduling paradigm, which assigns each parallel task a set of dedicated cores, achieves good theoretical bounds by ensuring exclusive use of processing resources to reduce interferences. However, because cores share the last-level cache and memory bandwidth resources, in practice tasks may still interfere with each other despite executing on dedicated cores. To tackle this issue, this work presents a holistic resource allocation framework for parallel real-time tasks under federated scheduling. Under the proposed framework, in addition to dedicated cores, each parallel task is also assigned with dedicated cache and memory bandwidth resources. This work also shows the study of the characteristics of parallel tasks upon different resource allocations following a measurement-based approach and proposes a technique to handle the challenge of tremendous profiling for all resource allocation combinations under this approach. Further, it proposes a holistic resource allocation algorithm that well balances the allocation between different resources to achieve good schedulability. Additionally, this work provides a full implementation of the framework by extending the federated scheduling system with Intel's Cache Allocation Technology and MemGuard. It also demonstrates the practicality of the proposed framework via extensive numerical evaluations and empirical experiments using real benchmark programs. In the end, the discussion about the application of risk measures for real-time scheduling is given for future work.

COMPUTATION OF RISK MEASURES IN FINANCE AND PARALLEL REAL-TIME SCHEDULING

by
Yajuan Li

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

August 2022

APPROVAL PAGE

COMPUTATION OF RISK MEASURES IN FINANCE AND PARALLEL REAL-TIME SCHEDULING

Yajuan Li

Jing Li, Dissertation Co-advisor — Date
Assistant Professor of Computer Science, NJIT

Marvin K. Nakayama, Dissertation Co-advisor — Date
Professor of Computer Science, NJIT

James M. Calvin, Committee Member — Date
Professor of Computer Science, NJIT

Pan Xu, Committee Member — Date
Assistant Professor of Computer Science, NJIT

Wenge Guo, Committee Member — Date
Associate Professor of Mathematical Sciences, NJIT

BIOGRAPHICAL SKETCH

Author: Yajuan Li

Degree: Doctor of Philosophy

Date: August 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2022
- Bachelor of Science in Information Management and Information Systems,
  Beijing Forestry University, Beijing, P.R. China, 2017

Major: Computer Science

Presentations and Publications:

Yajuan Li, Zachary T. Kaplan, and Marvin K. Nakayama, "Monte Carlo Methods for Economic Capital," *INFORMS Journal on Computing (in submission)*, 2022.

Lanshun Nie, Chenghao Fan, Shuang Lin, Li Zhang, Yajuan Li, and Jing Li, "Holistic Resource Allocation under Federated Scheduling for Parallel Real-Time Tasks," *ACM Trans. Embedd. Comput. Syst*, vol. 1, No. 1, Article 1, 2022.

Marvin K. Nakayama, Zachary T. Kaplan, Yajuan Li, Bruno Tuffin, and Pierre L'Ecuyer "Quantile Estimation via a Combination of Conditional Monte Carlo and Randomized Quasi-monte Carlo," *Winter Simulation Conference*, pp. 301–312, 2020.

Zachary T. Kaplan, Yajuan Li, Marvin K. Nakayama, and Bruno Tuffin, "Randomized Quasi-monte Carlo for Quantile Estimation," *Winter Simulation Conference*, pp. 428-439, 2019.

Zachary T. Kaplan, Yajuan Li, and Marvin K. Nakayama, "Monte Carlo Methods for Economic Capital," *Winter Simulation Conference*, pp. 1754-1765, 2018.

Where there is a will, there is a way.

Albert Einstein

ACKNOWLEDGMENT

I really appreciate my advisors Jing Li and Marvin K. Nakayama's responsible guidance and precious suggestions in my research work and personal life.

I would also like to thank my committee members: James M. Calvin, Pan Xu, and Wenge Guo for attending my proposal presentation and dissertation defense.

Thanks professor Marvin K. Nakayama for funding me with his National Science Foundation funding during 2017 fall–2019 fall. Thanks professor Jing Li for funding me with her university start-up funding during 2019 fall–2021 fall.

Finally, thank my beloved husband, Mr. Deng, for supporting me in everything, which greatly encouraged me in everyday life. Also, thank my family for taking care of me which gives me so much power.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

In this chapter, we will give the introduction of the following four chapters' topics, together with their contributions.

1.1 Quantile and Randomized Quasi-Monte Carlo

Quantile yields valuable insights in applications such as risk management, where answers to important questions lie in modeling the tails of the distribution. For a continuous random variable Y with cumulative distribution function (CDF) F, the p-quantile ($0 < p < 1$) is the smallest constant ξ such that $P(Y \leq \xi) = p$; i.e., $\xi = F^{-1}(p)$. For example, the median is the 0.5-quantile, also known as the 50th percentile. In finance, a quantile is called a *value-at-risk* (e.g., see [58]), which is often used to specify appropriate capital levels. Nuclear engineers employ a 0.95-quantile in *probabilistic safety assessments* (PSAs) of nuclear power plants. When a PSA is performed using *Monte Carlo* (MC), the U.S. Nuclear Regulatory Commission (NRC) requires accounting for the resulting sampling error; e.g., see [130], Section 3.2 of [128], and Section 24.9 of [131]. This can be accomplished by providing a *confidence interval* (CI) for ξ. Risk management aims to protect a financial institution from future uncertainties. For example, risk managers employ various *risk measures*, such as *quantile*, (e.g., Section 2.3 of [91]) to help determine appropriate levels of capital needed to be able to absorb (with high probability) large unexpected losses in credit portfolios comprising loans, bonds, and other financial instruments subject to default.

The typical MC approach to estimate ξ first estimates the CDF F, and then inverts the estimated CDF to obtain a quantile estimator; e.g., see Section 2.3 of [117]. Suppose a response Y can be generated from $d \geq 1$ independent and identically distributed (i.i.d.) uniforms on $[0,1)$. Then for a specified sample size $n \geq 1$, we can form an MC estimator of F by drawing a *random* sample of n independent uniformly distributed points from the unit

hypercube $[0,1)^d$; transforming each uniform vector into an observation of the response Y; and computing the empirical distribution function of the n values of Y.

In contrast, *quasi-Monte Carlo* (QMC) evaluates the response function at a *deterministic* set of n points that are carefully placed so that they more evenly cover $[0,1)^d$ than a typical random sample; see [99] and [79] for overviews of QMC. A QMC estimator of a mean can have a faster rate of convergence than the corresponding MC estimator. But providing explicit error bounds for a QMC estimator can be quite difficult.

Randomized QMC (RQMC) provides a way of producing computable error bounds. RQMC randomizes a QMC point set $r \geq 2$ independent times, where an estimator is computed from each randomization. Taking the sample mean and sample variance across the r randomizations, one can then compute a CI. RQMC has been implemented in various ways, including random shift modulo 1 [23, 124], random digital shift [80], and scrambling [104]. Previous work on applying QMC or RQMC for quantile estimation includes [109], [56], [68], and [48], but these works do not consider the problem of constructing explicit error bounds, as one wants for a nuclear PSA.

For an RQMC estimator of a mean when using a digital net with full nested scrambling ([104, 106, 105, 107]), [87] establishes a central limit theorem (CLT) with a normally distributed limit as the size m of the point set grows large, but the nested scrambling can be computationally costly. For other ways of implementing RQMC, as m gets large, an estimator computed from a randomization of the point set may not satisfy a CLT with a Gaussian limit. For example, for estimators based on randomly shifting (modulo 1) a lattice, [76] analyze the limiting distribution, which they generally find to be nonnormal, and their Figures 15 and 19 show histograms displaying distinct asymmetry and/or multiple modes. Thus, for a CI based on RQMC to be asymptotically valid, we may need the number r of randomizations to grow to infinity, which is the large-sample setting we now consider. In RQMC practice, though, it is common to choose r as not too large,

e.g., $r = 30$, motivated by a common rule of thumb for when the asymptotics of a CLT roughly start holding.

Contribution Chapter 2 examines asymptotic properties of two RQMC estimators of a quantile, where r grows large but the size m of the low-discrepancy point set remains fixed. In one approach, for each randomized point set, we compute a CDF estimator, which is inverted to obtain a quantile estimator. We then compute the sample average of the quantile estimators across the r randomizations to obtain the final quantile estimator. Because quantile estimators are generally biased, this estimator does *not* converge to the true quantile in our asymptotic regime, which keeps m fixed. This is in contrast to the corresponding RQMC estimator of a mean, which does converge to the true mean in this large-sample framework.

Our second quantile estimator instead computes a single CDF estimator using the responses from *all* randomizations, and then inverts the overall CDF estimator to obtain a single quantile estimator. We show that this RQMC quantile estimator, even though it is biased for fixed r and m, does converge to the true quantile as r grows large with m fixed. This RQMC quantile estimator also satisfies a [12] representation and a CLT with a Gaussian limit. We further provide numerical results comparing the two RQMC quantile estimators, along with MC estimators, as either r or m grows large, with the other fixed.

1.2 RQMC and Conditional Monte Carlo

In Chapter 3, we do the quantile estimation by combining two techniques. The first one, Conditional Monte Carlo (CMC), replaces the empirical CDF by an average of n realizations of the CDF of Y conditional on partial information that does not completely reveal Y, but from which we can easily compute its conditional CDF. It is often possible to do that in a way that the resulting (average) conditional CDF is continuous and smooth.

The second technique replaces the independent samples of Y by RQMC samples. More specifically, we assume that to simulate Y, we generate a vector $\mathbf{U}$ of d independent

random numbers uniform over $(0,1)$ that drive the simulation, where $d < \infty$ is fixed, and Y can be computed as a function of these random numbers. Many MC simulations work that way, sometimes with a random (and bounded) d, but this is often not a problem [73, 70]. RQMC replaces the n independent realizations of $\mathbf{U}$ by a set of n RQMC points in d dimensions. These points are not independent, but marginally, each of them has the uniform distribution over $(0,1)^d$, so each RQMC realization of Y has the correct distribution, and the n points together cover the unit cube more evenly (in some sense) than independent random points. If the CMC is designed so that the CDF estimator becomes a smooth function of $\mathbf{U}$ in the neighborhood of ξ_p, then the combination with RQMC can be much more effective, because the variance of the CDF estimator can then be reduced significantly, and may converge at a faster rate than the canonical one.

Because we assumed that $d < \infty$ is fixed, our paper does not cover infinite-dimensional problems, as can arise, e.g., when studying steady-state behavior or when some variates are generated via acceptance-rejection (because the number of rejections before acceptance is unbounded). But there are constructions that can dynamically add new coordinates to an RQMC point set, or pad the first QMC-generated coordinates by MC (pseudo-random) ones; e.g., see [100].

Prior work on applying QMC and RQMC for quantile estimation include [109], [56], [68], [48], and [61], although none of these papers combined it with CMC. Using CMC alone (without RQMC) for quantile estimation was proposed and studied in [95] and further discussed in [8]. Combining CMC with RQMC for *density* estimation is studied in [78]. These authors mention a possible application to quantile estimation, but leave its study for future work.

Contribution The standard estimator takes the p-quantile of the empirical distribution of independent observations obtained by MC. To get an improvement, Chapter 3 uses CMC to obtain a smoother estimate of the distribution function, and we combine this with RQMC to

further reduce the variance. The result is a much more accurate quantile estimator, whose mean square error can converge even faster than the canonical rate of $O(1/n)$.

1.3 Economic Capital and Measure-Specific Importance Sampling

Chapter 4 studies the *economic capital* (EC) $\eta = \xi - \mu$, the difference between the p-quantile ξ and the mean loss μ, where $p \approx 1$; see [66, p. 5], [88, Section 2.4], and [116, p. 194]. Also called the *credit* [59, p. 595], *relative* [57, p. 108] or *mean-adjusted VaR* [91, p. 300], the EC is used to determine capital needed to cover unexpected losses with high probability. Indeed, [24], p. 63, appears to employ EC with $p = 0.999$: "In line with our economic capital framework, economic capital for credit risk is set at a level to absorb with a probability of 99.9% very severe aggregate unexpected losses within one year. Our economic capital for credit risk is derived from the loss distribution of a portfolio via Monte Carlo Simulation of correlated rating migrations." (The bank used $p = 0.9998$ before 2017; see [24, p. 46].)

Monte Carlo simulation with *simple random sampling* (SRS) may produce noisy EC estimates because the rarity of extreme losses makes estimating ξ with $p \approx 1$ difficult. This motivates applying *variance-reduction techniques* (VRTs), such as *importance sampling* (IS); e.g., see Chapters V and VI of [9] and Chapter 4 of [38] for overviews. [40] develop IS methods for estimation of a tail probability of multifactor credit risk models using a Gaussian copula to model dependencies of default events among obligors (e.g., corporations to which the bank extended loans). [15] extend the IS methods to incorporate dependencies with non-Gaussian copulas.

While IS can be effective in reducing the variance of estimators of tail probabilities and extreme quantiles, it may produce worse estimators of the mean loss, the other component of the EC. An IS technique designed to work well for estimating an extreme quantile typically samples more in the tail of interest and less around the mean, degrading the mean's estimator. This motivates separately estimating the quantile and the mean

via different simulation techniques. In particular, we use IS for the estimation of the quantile and independently apply SRS for estimating the mean. [42] call this approach *measure-specific importance sampling* (MSIS), which they employ to separately estimate the numerator and denominator in a ratio of means in which only one mean corresponds to a rare event. We also consider two more methods that combine IS and SRS in other ways. One applies *IS with a defensive mixture* (ISDM), as developed by [50], in which the IS distribution is a mixture of a new distribution and the original one. The other approach we call a *double estimator* (DE), which utilizes both IS and SRS to estimate both ξ and μ.

We derive large-sample theory for the EC estimators, establishing a *central limit theorem* (CLT) for each. When the loss Y is the sum of m independent and identically distributed (i.i.d.) random variables, we also provide an analytical comparison of the estimators of η, ξ, and μ, in an asymptotic regime where $m \to \infty$ and the quantile level p simultaneously approaches 1. Originally developed by [41] to analyze SRS and IS estimators of quantiles, this limiting framework also has practical relevance for studying EC: bank portfolios can easily be exposed to thousands or even tens of thousands of obligors, and extreme quantiles are used in industry. Our theoretical analysis shows that MSIS can outperform the other methods to estimate η, which we additionally verify numerically. Through simulation experiments with a more complex portfolio credit risk model (PCRM), we further demonstrate the benefits when $p \approx 1$ of applying MSIS over the other approaches for the estimation of η. For the models in [40] and [15], calculating μ may not require simulation because of the tractability of their models. But more complicated stochastic models may preclude analytically evaluating the mean loss.

Contribution In Chapter 4, we analyze five different estimators of the EC. Although many of the techniques have been previously applied successfully to study problems arising in operations research and management science, some may not have been used before in a finance context. Our theoretical asymptotic analysis of the i.i.d. sum model (Section

4.5) provides insights into the behavior of the methods observed in experiments with the more complicated PCRM (with dependent obligors) in Section 4.6. While the analysis of SRS and IS quantile estimators in [41] covers only part (i.e., the numerator) of the asymptotic variances, we extend the theory by employing some approximations to study the entire asymptotic variances (both the numerators and denominators), providing a fuller understanding of the methods. Furthermore, our analysis examines the other methods for estimating quantiles and also studies all five estimators of the EC. The behavior of the EC estimators on the i.i.d. sum model carries over to the simulation experiments with the PCRM (Section 4.6.2): MSIS and ISDM greatly outperform the other methods, with MSIS slightly better than ISDM.

1.4 Federated Scheduling and Holistic Resource Allocation

Because workload consolidation can effectively reduce the energy consumption, wiring weight, hardware costs, and software complexity, recently there is a technology trend of consolidating even more applications and services onto shared hardware for embedded systems and edge servers. On the hardware side, there is a rapidly increasing penetration of multi-core/many-core CPUs into systems, as well as an increased sharing of common resources by computing units (e.g., memory bus and last-level cache). Moreover, the applications running in embedded systems and edge servers today have increasingly high computation needs and stringent timing constraints. For example, an edge server needs to provide real-time responsiveness to various applications, such as augmented reality, video analytics and traffic light controls, that offload computation to the shared edge [119, 115]. These technology trends mean that: (1) the parallel execution is critical for satisfying the high computation needs and meeting the stringent real-time constraints of applications; and (2) the execution of applications is more unpredictable due to computing units sharing resources like cache and memory bus.

Among the different scheduling policies for executing parallel real-time tasks upon multi-core platforms, the federated scheduling paradigm [82, 54, 127, 3, 14, 13, 123] has attracted a lot of attention due to its good theoretical bounds and empirical advantages. The key idea of the federated scheduling paradigm is to allocate a set of dedicated cores to each task that needs to run in parallel on multiple cores to meet its deadline and force the remaining tasks to execute serially on the remaining cores under some classical multiprocessor scheduling algorithms. Because each task that runs in parallel has its dedicated cores for execution, there is no preemption, migration, and interference on its cores caused by other tasks. In addition to reducing practical overheads, each of such tasks can be analyzed in isolation, which significantly reduces the pessimism of analyzing the schedulability of complex parallel real-time tasks. Thus, federated scheduling achieves the best performance bounds compared to other classical algorithms, such as global earliest deadline first and global rate monotonic scheduling.

However, in today's multicore hardware, cores share the last-level cache and memory bandwidth resources, so tasks may interfere with each other despite executing on dedicated cores. The interferences due to cache and memory bandwidth contention can be even more severe for embedded platforms or edge servers, where the computing power and cache/memory space are limited.

Chapter 5 aims to address the pressing demand for *parallel real-time scheduling over multicore platforms with shared cache and memory bus*. Specifically, we focus on platforms with multicore processors with L1 (and L2) caches private to each core and shared Last Level Cache (LLC), which is connected via a shared memory bus to the shared Direct Random Access Memory (DRAM). In this work, we take a measurement-based approach and limit our attention to addressing the contention in the shared LLC and memory bus for parallel real-time tasks. Note that there are other sources of interference on modern platforms, such as Miss Status Holding Registers (MSHRs) [132], DRAM bank conflicts [139, 63], and DRAM controller [112, 121], as well as the contention in

software systems like blocking due to shared internal kernel data structures [144]. Some of these interferences (e.g., MSHRs contention) can be mostly incorporated into the measured worst-case execution times by co-running the task with specially designed interfering workloads during the profiling, while some have been mitigated by various mechanisms. We leave the integration of the proposed strategy and the orthogonal mitigation mechanisms to other resources, such as DRAM bank-level partitioning, as future work.

Contribution In Chapter 5, we first study the characteristics of parallel tasks upon different resource allocations following a measurement-based approach. Since each task can be allocated with different numbers of cores, cache partitions, and memory bandwidth partitions, profiling the worst-case execution times for all the thousands of combinations of resource allocation can take a tremendous amount of time. To address this issue, we propose to perform the measurement only for a small number of combinations and apply a non-linear regression to obtain estimations for the other combinations.

Next, we present a holistic cache and memory bandwidth resource allocation strategy for parallel real-time tasks under federated scheduling. In addition to dedicated cores, each parallel task is also assigned with dedicated cache and memory bandwidth resources to reduce resource interferences between tasks. We leverage the insights from the heuristic resource allocation strategy CaM[136] for allocating cache and memory bandwidth for sequential tasks and extend the federated scheduling system using Intel's Cache Allocation Technology and MemGuard for allocating cache and memory bandwidth. To balance well the allocation between different resources and achieve good schedulability, we develop a mixed-integer nonlinear programming (MINLP) formulation that can optimally solve this problem. Moreover, we propose a heuristic-based greedy algorithm that has good schedulability and short running times that are orders of magnitude faster than solving the MINLP. Additionally, we provide a full implementation of our framework by extending the federated scheduling system with Intel's Cache Allocation Technology and MemGuard.

Finally, we demonstrate the practicality of our proposed framework via extensive numerical evaluations and empirical experiments using real benchmark programs.

CHAPTER 2

RANDOMIZED QUASI-MONTE CARLO FOR QUANTILE ESTIMATION

This Chapter considered two different randomized quasi-monte carlo (RQMC) quantile estimators: the first estimator converges to the wrong value as randomization number goes to infinity, in contrast, our other RQMC quantile estimator does converge to the desired value of quantile.

The rest of this chapter unfolds as follows. Section 2.1 describes the basic mathematical problem. In Sections 2.2 and 2.3, we review how to estimate a quantile using MC and QMC, respectively. Sections 2.4 and 2.5 develop our two RQMC quantile estimators. We provide numerical results in Section 2.6. All proofs will appear in a follow-up paper.

2.1 Mathematical Background

For a given (deterministic) function $w_Y : [0,1)^d \to \Re$ with fixed integer $d \geq 1$, let

$$Y = w_Y(U_1, U_2, \ldots, U_d) = w_Y(\mathbf{U}) \tag{2.1}$$

where $U_1, U_2, \ldots, U_d$, are i.i.d. $U[0,1)$ (i.e., uniform on the interval $[0,1)$) random variables, so $\mathbf{U} = (U_1, U_2, \ldots, U_d) \sim U[0,1)^d$. We call w_Y a *response function*, which may represent a simulation program that produces a response Y using d i.i.d. uniforms as input. The function w_Y can be quite complicated, first converting $\mathbf{U} \sim U[0,1)^d$ into a random vector $\mathbf{Z}$ with non-identically distributed components having a dependence structure, and then performing computations using $\mathbf{Z}$, to finally produce Y. Let F be the CDF of Y, which we assume cannot be computed analytically nor numerically. For each $y \in \Re$, we have that

$$F(y) = P(Y \leq y) = P(w_Y(\mathbf{U}) \leq y) = \int_{[0,1)^d} I(w_Y(\mathbf{u}) \leq y)\, \mathrm{d}\mathbf{u}, \tag{2.2}$$

where $I(\cdot)$ denotes the *indicator function*, which equals 1 (resp., 0) when its argument is true (resp., false).

For a fixed value $0 < p < 1$, define $\xi \equiv \xi_p = F^{-1}(p) \equiv \inf\{y : F(y) \geq p\}$, which is the *p-quantile* of F (equivalently, of Y). Thus, in the case that F is continuous, exactly p of the mass of F lies below ξ. Let f denote the derivative (when it exists) of F, and we will assume throughout that $f(\xi) > 0$, which ensures that $y = \xi$ is the unique solution of the equation $F(y) = p$.

The goal is to estimate ξ using some form of Monte Carlo or quasi-Monte Carlo. The general approach we will follow to estimate $\xi = F^{-1}(p)$ is to first estimate the CDF F and then invert the estimated CDF to obtain a quantile estimator. We further want to provide a measure of the error of our quantile estimator.

We next motivate the problem and illustrate the notation in the following example.

Example 1 Consider a system experiencing a random load L with a random capacity C to withstand the load. The system fails when $L \geq C$, so $Y \equiv C - L$ is the system's *safety margin*, which has CDF F. An example is a nuclear power plant undergoing a hypothesized accident, as studied in [28] and [118], where L denotes the peak cladding temperature (PCT) during the postulated accident and C is the temperature at which the cladding material suffers damage. It is reasonable to consider the PCT as a random variable because it depends on unforeseen aspects of the events (e.g., time and size of a pipe break) during the accident, and the capacity C may be unknown because of the variability of the cladding's material properties, which are modeled as random variables.

In Equation (2.2), we can think of the function w_Y as follows. It first takes d i.i.d. uniforms as input, transforming them into an observation of (L,C), possibly with some dependence structure. Then w_Y outputs $Y = C - L$.

Let $\theta = P(Y \leq 0)$, which is the failure probability, and a regulator may specify that θ must be less than a given threshold θ_0, e.g., $\theta_0 = 0.05$. The requirement that $\theta < \theta_0$ can be equivalently reformulated in terms of a quantile: the θ_0-quantile ξ of Y must satisfy $\xi > 0$.

2.2 Monte Carlo

We now describe how to apply MC to estimate ξ. Fix a sample size $n \geq 2$, and generate a sample of n independent random vectors $\mathbf{U}_i$, $i = 1, 2, \ldots, n$, where each $\mathbf{U}_i = (U_{i,1}, U_{i,2}, \ldots, U_{i,d}) \sim U[0,1)^d$. For each $i = 1, 2, \ldots, n$, define $Y_i = w_Y(\mathbf{U}_i)$, so $Y_1, Y_2, \ldots, Y_n$ is a sample of n i.i.d. copies of Y, with each $Y_i \sim F$ by Equation (2.1). Then we define the MC estimator of F as the *empirical distribution function*

$$\widehat{F}_{\mathrm{MC},n}(y) = \frac{1}{n} \sum_{i=1}^{n} I(Y_i \leq y). \tag{2.3}$$

A natural estimator of $\xi = F^{-1}(p)$ is the *MC quantile estimator*

$$\widehat{\xi}_{\mathrm{MC},n} = \widehat{F}^{-1}_{\mathrm{MC},n}(p), \tag{2.4}$$

which can be computed through *order statistics*. Specifically, let $Y_{1:n} \leq Y_{2:n} \leq \cdots \leq Y_{n:n}$ be the ordered values of the sample $Y_1, Y_2, \ldots, Y_n$. Let $\lceil \cdot \rceil$ denote the ceiling function, and we have that

$$\widehat{\xi}_{\mathrm{MC},n} = Y_{\lceil np \rceil :n}. \tag{2.5}$$

2.2.1 Large-Sample Properties of MC Quantile Estimator

A *Bahadur representation* [12], described next, provides a useful approach for analyzing the large-sample properties of $\widehat{\xi}_{\mathrm{MC},n}$. For n sufficiently large, there exists a neighborhood $\mathcal{N}_n$ of ξ such that

$$\widehat{F}_{\mathrm{MC},n}(y) \approx \widehat{F}_{\mathrm{MC},n}(\xi) + F(y) - F(\xi) \quad \text{uniformly for } y \text{ in } \mathcal{N}_n,$$

and $\mathcal{N}_n$ contains $\widehat{\xi}_{\mathrm{MC},n}$ with probability 1. Thus, because $\widehat{\xi}_{\mathrm{MC},n} = \widehat{F}^{-1}_{\mathrm{MC},n}(p)$, we have that

$$p \approx \widehat{F}_{\mathrm{MC},n}(\widehat{\xi}_{\mathrm{MC},n}) \approx \widehat{F}_{\mathrm{MC},n}(\xi) + F(\widehat{\xi}_{\mathrm{MC},n}) - F(\xi) \approx \widehat{F}_{\mathrm{MC},n}(\xi) + f(\xi)(\widehat{\xi}_{\mathrm{MC},n} - \xi),$$

where the last step follows from a first-order Taylor approximation. Under our assumption from Section 2.1 that $f(\xi) > 0$, rearranging terms leads to

$$\widehat{\xi}_{\mathrm{MC},n} \approx \xi + \frac{p - \widehat{F}_{\mathrm{MC},n}(\xi)}{f(\xi)},$$

so the quantile estimator roughly equals the true quantile plus a linear transformation of a CDF estimator evaluated at ξ.

Reference [12] formalizes the above discussion. Specifically, if $f(\xi) > 0$, then

$$\widehat{\xi}_{\mathrm{MC},n} = \xi + \frac{p - \widehat{F}_{\mathrm{MC},n}(\xi)}{f(\xi)} + R_n, \tag{2.6}$$

$$\text{with } \sqrt{n}R_n \Rightarrow 0 \text{ as } n \to \infty, \tag{2.7}$$

where $\Rightarrow$ denotes convergence in distribution (e.g., Section 25 of [17]). We call (2.6)–(2.7) a (weak) *Bahadur representation*. Under the additional assumption that F is twice differentiable at ξ, [12] actually proves a stronger result than (2.7), namely that R_n vanishes at rate $O(n^{-3/4}\log n)$ almost surely (a.s.); see Section 2.5 of [117] for refinements.

The Bahadur representation implies that the MC quantile estimator satisfies a CLT. From (2.6), we have

$$\sqrt{n}\left[\widehat{\xi}_{\mathrm{MC},n} - \xi\right] = \frac{\sqrt{n}}{f(\xi)}\left[p - \widehat{F}_{\mathrm{MC},n}(\xi)\right] + \sqrt{n}R_n. \tag{2.8}$$

By Equation (2.3), $\widehat{F}_{\mathrm{MC},n}(\xi)$ averages i.i.d. copies of $I(Y \le \xi)$, which has mean $E[I(Y \le \xi)] = p$ and variance $\psi^2_{\mathrm{MC}} \equiv \mathrm{Var}[I(Y \le \xi)] = p(1-p)$. Thus, the *ordinary CLT* (e.g., Theorem 27.1 of [17]) ensures

$$\sqrt{n}\left[p - \widehat{F}_{\mathrm{MC},n}(\xi)\right] \Rightarrow N(0, \psi^2_{\mathrm{MC}}) \quad \text{as } n \to \infty, \tag{2.9}$$

where $N(a, b^2)$ is a normal random variable with mean a and variance b^2. Hence, using (2.9) and (2.7) in (2.8), and applying Slutsky's theorem (e.g., p. 19 of [117]), we get that

$$\sqrt{n}\left[\widehat{\xi}_{\mathrm{MC},n} - \xi\right] \Rightarrow \frac{1}{f(\xi)}N(0, \psi^2_{\mathrm{MC}}) + 0 \stackrel{\mathscr{D}}{=} N(0, \kappa^2_{\mathrm{SRS}}) \quad \text{as } n \to \infty, \tag{2.10}$$

where $\stackrel{\mathscr{D}}{=}$ denotes equality in distribution, and

$$\kappa^2_{\mathrm{SRS}} \equiv \frac{\psi^2_{\mathrm{MC}}}{f^2(\xi)} = \frac{p(1-p)}{f^2(\xi)}. \tag{2.11}$$

Therefore, even though $\widehat{\xi}_{\mathrm{MC},n}$ is *not* a sample average, it still obeys a CLT because the Bahadur representation shows that the large-sample asymptotics of $\widehat{\xi}_{\mathrm{MC},n}$ can be well-approximated by those of $\widehat{F}_{\mathrm{MC},n}(\xi)$,which *is* a sample average satisfying the CLT in Equation (2.9).

One measure of the error of a Monte Carlo estimator is its (root) mean squared error ((R)MSE). For our MC quantile estimator $\widehat{\xi}_{\mathrm{MC},n}$ in (2.4), Theorem 2 of [10] shows that

$$\mathrm{MSE}\left[\widehat{\xi}_{\mathrm{MC},n}\right] = E\left[(\widehat{\xi}_{\mathrm{MC},n} - \xi)^2\right] = n^{-1}\kappa^2_{\mathrm{SRS}} + o(n^{-1}) = O(n^{-1})$$

as $n \to \infty$, where for two functions $g_1(n)$ and $g_2(n)$, we write that $g_1(n) = O(g_2(n))$ as $n \to \infty$ if there exists a constant c such that $|g_1(n)| \leq c|g_2(n)|$ for all n sufficiently large, and $g_1(n) = o(g_2(n))$ means that $g_1(n)/g_2(n) \to 0$ as $n \to \infty$. Thus, although $\widehat{\xi}_{\mathrm{MC},n}$ is generally biased, its MSE is dominated by its asymptotic variance κ^2_{SRS} from (2.11); also see Lemma 1 of [10]. We then see that

$$\mathrm{RMSE}\left[\widehat{\xi}_{\mathrm{MC},n}\right] = n^{-1/2}\kappa_{\mathrm{SRS}} + o(n^{-1/2}) = O(n^{-1/2}) \tag{2.12}$$

as $n \to \infty$, which provides a measure of the rate of convergence of the MC quantile estimator.

Another way of describing the error in $\widehat{\xi}_{\mathrm{MC},n}$ is through a confidence interval. We can unfold the CLT Equation (2.10) to obtain an asymptotic β-level ($0 < \beta < 1$) two-sided CI for ξ as $J'_{\mathrm{MC},n} \equiv [\widehat{\xi}_{\mathrm{MC},n} \pm z_{1-(1-\beta)/2}\kappa_{\mathrm{SRS}}/\sqrt{n}]$, where $z_q = \Phi^{-1}(q)$ for $0 < q < 1$ and Φ is the $N(0,1)$ CDF (e.g., $z_{0.95} = 1.96$). However, this CI is not directly implementable because $f(\xi)$ in κ^2_{SRS} of (2.11) is typically unknown. But it is possible (e.g., see [18]) to construct a *consistent* estimator $\widehat{\tau}^2_{\mathrm{MC},n}$ of κ^2_{SRS}; i.e., $\widehat{\tau}^2_{\mathrm{MC},n} \Rightarrow \kappa^2_{\mathrm{SRS}}$ as $n \to \infty$. We can then

obtain a large-sample β-level two-sided CI for ξ as

$$J_{\mathrm{MC},n} \equiv \left[\widehat{\xi}_{\mathrm{MC},n} \pm z_{1-(1-\beta)/2}\widehat{\tau}_{\mathrm{MC},n}/\sqrt{n}\right],$$

which is *asymptotically valid* in the sense that $\lim_{n\to\infty} P(\xi \in J_{\mathrm{MC},n}) = \beta$, or equivalently,

$$P\left(|\widehat{\xi}_{\mathrm{MC},n} - \xi| \le z_{1-(1-\beta)/2}\widehat{\tau}_{\mathrm{MC},n}/\sqrt{n}\right) \to \beta, \quad \text{as } n \to \infty.$$

As a consequence, we have that $|\widehat{\xi}_{\mathrm{MC},n} - \xi| = O_p(n^{-1/2})$ as $n \to \infty$, where the notation $X_n = O_p(a_n)$ for a sequence of random variables X_n, $n \ge 1$, and constants a_n, $n \ge 1$, means that X_n/a_n is bounded in probability (Section 1.2.5 of [117]).

2.3 Quasi-Monte Carlo

Rather than estimating ξ with *random sampling* as in Monte Carlo, QMC instead evaluates the response function at carefully placed *deterministic* points in $[0,1)^d$, which are chosen to be more evenly dispersed over $[0,1)^d$ than a typical random sample of i.i.d. uniforms. Let $\mathscr{P}_n = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ be a *low-discrepancy point set* of size n, where each $\mathbf{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,d}) \in [0,1)^d$. Such a $\mathscr{P}_n$ can be constructed deterministically as a *lattice* [120] or a *digital net*, including ones designed by Halton, Faure, Sobol', and Niederreiter; see Chapters 3–5 of [99] or Chapter 5 of [79] for an overview.

The QMC estimator of $F(y)$ in Equation (2.2) is

$$\widehat{F}_{\mathrm{QMC},n}(y) = \frac{1}{n}\sum_{i=1}^{n} I(w_Y(\mathbf{u}_i) \le y).$$

We call $\widehat{F}_{\mathrm{QMC},n}(\cdot)$ the *QMC CDF estimator*, and we invert $\widehat{F}_{\mathrm{QMC},n}$ to obtain the *QMC quantile estimator*

$$\widehat{\xi}_{\mathrm{QMC},n} = \widehat{F}^{-1}_{\mathrm{QMC},n}(p). \tag{2.13}$$

Just as for the MC p-quantile estimator in Equation (2.5), we can compute $\widehat{\xi}_{\mathrm{QMC},n}$ in (2.13) by sorting $w_Y(\mathbf{u}_i)$, $i = 1, 2, \ldots, n$, in ascending order, and setting $\widehat{\xi}_{\mathrm{QMC},n}$ equal to the $\lceil np \rceil$-th smallest one.

Recall that by Equation (2.12), the RMSE of the MC quantile estimator converges at rate $O(n^{-1/2})$, where n is the sample size, and we would like to provide analogous (deterministic) error bounds for $|\widehat{\xi}_{\mathrm{QMC},n} - \xi|$. One approach is to try the following. For the moment, suppose that we are interested in computing the integral

$$\gamma \equiv \int_{[0,1)^d} h(\mathbf{u})\, \mathrm{d}\mathbf{u} \tag{2.14}$$

for some integrand $h : [0,1)^d \to \Re$, and we estimate γ by the QMC estimator $(1/n)\sum_{i=1}^n h(\mathbf{u}_i)$, with low-discrepancy point set $\mathscr{P}_n = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$. Then the *Koksma-Hlawka inequality* states that

$$\left| \frac{1}{n} \sum_{i=1}^{n} h(\mathbf{u}_i) - \gamma \right| \leq D^*(\mathscr{P}_n) V_{\mathrm{HK}}(h), \tag{2.15}$$

where $D^*(\mathscr{P}_n)$ is the star-discrepancy of $\mathscr{P}_n$, which is a measure of the uniformity of $\mathscr{P}_n$ over $[0,1)^d$, and $V_{\mathrm{HK}}(h)$ is the Hardy-Krause (HK) variation of the integrand h, specifying its roughness; see Section 5.6 of [79] for details. Low-discrepancy point sets $\mathscr{P}_n$ often have $D^*(\mathscr{P}_n) = O((\log n)^v/n)$ for some constant $v > 0$ (e.g., $v = d-1$ or $v = d$) as $n \to \infty$. Hence, when the integrand h is sufficiently smooth so that $V_{\mathrm{HK}}(h) < \infty$, (2.15) implies that the deterministic rate at which the QMC integration error decreases is $O((\log n)^v/n)$ as $n \to \infty$, better than MC's rate of $O(n^{-1/2})$.

But there are several problems with this approach of trying to bound the QMC error. When estimating the CDF $F(y)$ in Equation (2.2), the integrand is $h_y(\mathbf{u}) = I(w_Y(\mathbf{u}) \leq y)$, which is discontinuous in $\mathbf{u}$ and typically has $V_{\mathrm{HK}}(h_y) = \infty$, so the upper bound in the Koksma-Hlawka inequality (2.15) is infinite. Even if the HK variation of the integrand h were finite, computing the bound in (2.15) is at least as difficult as computing the integral in (2.14), and the bound can be quite conservative, making (2.15) impractical. Moreover, the

bound in (2.15) for integrand h_y is for the QMC CDF estimator at y, not for $|\widehat{\xi}_{\mathrm{QMC},n} - \xi|$, which is what we are actually interested in.

2.4 One Approach of Randomized Quasi-Monte Carlo

Rather than trying to provide a *deterministic* error bound for the QMC quantile estimator, we can instead attempt to use RQMC to obtain a CI for ξ. For a given low-discrepancy point set, the basic idea is to randomize the set $r \geq 2$ independent times in a way that retains the low-discrepancy property for each randomization, and compute an estimator from each of the r independent randomizations. Then we can form a CI from the sample mean and sample variance across randomizations. For a fair comparison to the p-quantile estimator using MC in Equation (2.4) or via QMC in Equation (2.13), each of which is based on n evaluations of the response function w_Y in Equation (2.1), we also want to apply RQMC using the same total number n of function evaluations. We next describe details on how RQMC may be implemented.

Let $r \geq 2$ be the number of randomizations to use for RQMC, and let $\mathscr{P}_m = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ be a low-discrepancy point set of size $m = n/r$, where each $\mathbf{u}_i \in [0,1)^d$, and we assume that n/r is an integer. For each $k = 1, 2, \ldots, r$, we want to perform a randomization of $\mathscr{P}_m$ to obtain another point set

$$\mathscr{P}_m^{(k)} = \{\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_m^{(k)}\}, \tag{2.16}$$

with each $\mathbf{X}_i^{(k)} = (X_{i,1}^{(k)}, X_{i,2}^{(k)}, \ldots X_{i,d}^{(k)})$, such that

$$\mathbf{X}_i^{(k)} \sim U[0,1)^d, \text{ for each } i = 1, 2, \ldots, m, \text{ and each } k = 1, 2, \ldots, r, \tag{2.17}$$

and

$$\mathscr{P}_m^{(1)}, \mathscr{P}_m^{(2)}, \ldots, \mathscr{P}_m^{(r)} \text{ are i.i.d.} \tag{2.18}$$

One simple way of constructing $\mathscr{P}_m^{(k)}$ in (2.16) satisfying (2.17) and (2.18) is through independent *random shifts*. First generate $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_r$ as r independent random vectors, where each $\mathbf{S}_k = (S_{k,1}, S_{k,2}, \ldots, S_{k,d}) \sim U[0,1)^d$. For each $k = 1, 2, \ldots, r$, we will shift (modulo 1) each point in the original set $\mathscr{P}_m$ by $\mathbf{S}_k$ to obtain $\mathscr{P}_m^{(k)}$. Specifically, for $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \Re^d$ and $\mathbf{y} = (y_1, y_2, \ldots, y_d) \in \Re^d$, define the operator $\oplus$ as $\mathbf{x} \oplus \mathbf{y} = ((x_1 + y_1) \bmod 1, (x_2 + y_2) \bmod 1, \ldots, (x_d + y_d) \bmod 1)$. For each $k = 1, 2, \ldots, r$, we then obtain a shifted point set $\mathscr{P}_m^{(k)}$ in Equation (2.16) with each $\mathbf{X}_i^{(k)} = \mathbf{u}_i \oplus \mathbf{S}_k$; i.e., each point in the original point set is shifted by the same random uniform $\mathbf{S}_k$. It is easy to show that each $\mathbf{u}_i \oplus \mathbf{S}_k \sim U[0,1)^d$, so Equation (2.17) holds. Each shifted point set $\mathscr{P}_m^{(k)}$ uses the same low-discrepancy point set $\mathscr{P}_m$ but a different random shift $\mathbf{S}_k$. The m points in any $\mathscr{P}_m^{(k)}$ will be stochastically dependent because they all share the same random shift $\mathbf{S}_k$. But the r shifted point sets $\mathscr{P}_m^{(k)}$, $k = 1, 2, \ldots, r$, are stochastically independent because $\mathbf{S}_k$, $k = 1, 2, \ldots, r$, are independent, thereby implying Equation (2.18).

When the original point set $\mathscr{P}_m$ is a lattice, each shifted point set $\mathscr{P}_m^{(k)}$ retains a lattice structure. But if $\mathscr{P}_m$ is a digital net, its random shift $\mathscr{P}_m^{(k)}$ may no longer be a digital net. In this latter case, we instead can apply *scrambling* to obtain each $\mathscr{P}_m^{(k)}$ satisfying Equations (2.17) and (2.18), where the scrambled point set still possesses the desirable properties of the original point set; see [105, 106].

In whatever way we obtain the r randomized point sets $\mathscr{P}_m^{(1)}, \mathscr{P}_m^{(2)}, \ldots, \mathscr{P}_m^{(r)}$ satisfying Equations (2.17) and (2.18), for each randomization $k = 1, 2, \ldots, r$, let

$$\widehat{F}_{\mathrm{RQMC},m,k}(y) = \frac{1}{m} \sum_{i=1}^{m} I(w_Y(\mathbf{X}_i^{(k)}) \leq y),$$

which is a CDF estimator computed from the randomized point set $\mathscr{P}_m^{(k)}$. Even though $\mathbf{X}_i^{(k)}$, $i = 1, 2, \ldots, m$, are stochastically dependent, we still have that $E[\widehat{F}_{\mathrm{RQMC},m,k}(y)] = F(y)$ for each y by Equation (2.17). We invert $\widehat{F}_{\mathrm{RQMC},m,k}$ to obtain

$$\widehat{\xi}_{\mathrm{RQMC},m,k} = \widehat{F}^{-1}_{\mathrm{RQMC},m,k}(p), \tag{2.19}$$

where $\widehat{\xi}_{\mathrm{RQMC},m,k}$, $k = 1, 2, \ldots, r$, are i.i.d. Then an RQMC estimator of ξ is

$$\overline{\xi}_{\mathrm{RQMC},m,r} = \frac{1}{r}\sum_{k=1}^{r} \widehat{\xi}_{\mathrm{RQMC},m,k}. \tag{2.20}$$

As noted previously at the end of Section 2.3, when estimating the CDF in Equation (2.2), the integrand $I(w_Y(\mathbf{u}) \le y)$ typically has infinite HK variation, making theoretical bounds of the form in Equation (2.15) not useful. But when instead estimating the integral in Equation (2.14) with an integrand $h(\mathbf{u})$ having finite HK variation, applying RQMC leads to certain benefits over MC and QMC. First, for the estimator from each randomization in RQMC, the rate at which its variance decreases is $O(m^{-2}(\log m)^d)$ as m grows, and even faster in some cases, as detailed by [80, 106, 125]. In contrast, the MC variance of a sample of size m decreases at rate $O(m^{-1})$. Compared to QMC, RQMC allows for practical estimation of the approximation error through a CLT. Moreover, even when the HK variation is infinite, making inequalities such as Equation (2.15) uninformative, numerical experiments show that the empirical behavior of RQMC's convergence rate can be substantially better than that of MC [126].

2.4.1 Large-Sample Properties of RQMC Quantile Estimator $\overline{\xi}_{\mathrm{RQMC},m,r}$

The RQMC quantile estimator $\overline{\xi}_{\mathrm{RQMC},m,r}$ in (2.20) satisfies the following CLT, where $(\widehat{F}_{\mathrm{RQMC},m}, \widehat{\xi}_{\mathrm{RQMC},m})$ denotes a generic copy of $(\widehat{F}_{\mathrm{RQMC},m,k}, \widehat{\xi}_{\mathrm{RQMC},m,k})$.

Proposition 1 If $\tau'^2_{\mathrm{RQMC},m} \equiv \mathrm{Var}[\widehat{\xi}_{\mathrm{RQMC},m}] < \infty$, then for fixed $m \ge 1$,

$$\sqrt{r}\left(\overline{\xi}_{\mathrm{RQMC},m,r} - E[\widehat{\xi}_{\mathrm{RQMC},m}]\right) \Rightarrow N(0, \tau'^2_{\mathrm{RQMC},m}) \quad \text{as } r \to \infty. \tag{2.21}$$

It is important to note that the centering constant on the left-hand side of CLT (2.21) is *not* the true p-quantile ξ but rather $E[\widehat{\xi}_{\mathrm{RQMC},m}]$. While $\widehat{F}_{\mathrm{RQMC},m}(y)$ is an unbiased estimator of $F(y)$ for each m and y, its inverse $\widehat{\xi}_{\mathrm{RQMC},m} = \widehat{F}^{-1}_{\mathrm{RQMC},m}(p)$ is typically a *biased* estimator of ξ because of the nonlinearity of the inversion operation. Because $\overline{\xi}_{\mathrm{RQMC},m,r}$ averages r

i.i.d. copies of $\widehat{\xi}_{\mathrm{RQMC},m}$, we see that

$$E[\overline{\xi}_{\mathrm{RQMC},m,r}] = E[\widehat{\xi}_{\mathrm{RQMC},m}] \neq \xi \tag{2.22}$$

in general for fixed m. In addition, for fixed m, we have that a.s.,

$$\lim_{r\to\infty} \overline{\xi}_{\mathrm{RQMC},m,r} = E[\widehat{\xi}_{\mathrm{RQMC},m}] \neq \xi \tag{2.23}$$

by the strong law of large numbers, so $\overline{\xi}_{\mathrm{RQMC},m,r}$ converges to the wrong value as $r \to \infty$ for fixed m.

Because $\mathrm{Bias}[\overline{\xi}_{\mathrm{RQMC},m,r}] = \mathrm{Bias}[\widehat{\xi}_{\mathrm{RQMC},m}] = E[\widehat{\xi}_{\mathrm{RQMC},m}] - \xi$, we have that

$$\begin{aligned}
\mathrm{MSE}\left[\overline{\xi}_{\mathrm{RQMC},m,r}\right] &= \left(\mathrm{Bias}\left[\overline{\xi}_{\mathrm{RQMC},m,r}\right]\right)^2 + \mathrm{Var}\left[\overline{\xi}_{\mathrm{RQMC},m,r}\right] \\
&= \left(\mathrm{Bias}\left[\widehat{\xi}_{\mathrm{RQMC},m}\right]\right)^2 + \frac{1}{r}\mathrm{Var}\left[\widehat{\xi}_{\mathrm{RQMC},m}\right].
\end{aligned} \tag{2.24}$$

Although the second term in (2.24) decreases at rate r^{-1} as $r \to \infty$, (2.22) implies that the first is nonzero and does not shrink for fixed m, so

$$\mathrm{RMSE}\left[\overline{\xi}_{\mathrm{RQMC},m,r}\right] = \mathrm{Bias}\left[\widehat{\xi}_{\mathrm{RQMC},m}\right] + o(1) = O(1) \tag{2.25}$$

as $r \to \infty$ with m fixed. Thus, the RMSE of $\overline{\xi}_{\mathrm{RQMC},m,r}$ does not converge to 0 as $r \to \infty$ for fixed m.

Moreover, suppose we unfold the CLT (2.21) to build a β-level CI for ξ as

$$J_{\mathrm{RQMC},m,r} = (\overline{\xi}_{\mathrm{RQMC},m,r} \pm z_{1-(1-\beta)/2}\widehat{\tau}'_{\mathrm{RQMC},m,r}/\sqrt{r}), \tag{2.26}$$

where $\widehat{\tau}'^2_{\mathrm{RQMC},m,r} = (1/(r-1))\sum_{k=1}^{r}[\widehat{\xi}_{\mathrm{RQMC},m,k}) - \overline{\xi}_{\mathrm{RQMC},m,r}]^2$ is a consistent estimator of $\tau'^2_{\mathrm{RQMC},m}$. Because the midpoint of $J_{\mathrm{RQMC},m,r}$ is the biased estimator $\overline{\xi}_{\mathrm{RQMC},m,r}$, the CI is centered at the wrong point on average, which can lead to poor coverage as $r \to \infty$ with m fixed. We can try to address this issue by also letting $m \to \infty$, but we would then need to determine the relative rates at which $m \to \infty$ and $r \to \infty$ to ensure a CLT still holds.

2.5 Another Approach of RQMC for Quantile Estimation

As we explained in Section 2.4.1, the RQMC quantile estimator $\overline{\xi}_{\mathrm{RQMC},m,r}$ in Equation (2.20) does not converge to ξ as $r \to \infty$ for fixed m. We next consider another RQMC estimator that, although biased, does converge in this setting.

Rather than compute a quantile estimator from each of the r randomizations, as in Equation (2.19), we instead construct a single overall CDF estimator from all r randomizations, and then invert this to obtain a single overall quantile estimator. Specifically, first define the CDF estimator based on all rm evaluations of the response function w_Y as

$$\widetilde{F}_{\mathrm{RQMC},m,r}(y) = \frac{1}{r}\sum_{k=1}^{r} \widehat{F}_{\mathrm{RQMC},m,k}(y) = \frac{1}{rm}\sum_{k=1}^{r}\sum_{i=1}^{m} I(w_Y(\mathbf{X}_i^{(k)}) \le y), \tag{2.27}$$

which we call the *overall CDF estimator*. We then invert this to obtain another RQMC quantile estimator

$$\widetilde{\xi}_{\mathrm{RQMC},m,r} = \widetilde{F}^{-1}_{\mathrm{RQMC},m,r}(p). \tag{2.28}$$

2.5.1 Large-Sample Properties of RQMC Quantile Estimator $\widetilde{\xi}_{\mathrm{RQMC},m,r}$

Because $\widehat{F}_{\mathrm{RQMC},m,k}$, $k = 1,2,\ldots,r$, are i.i.d., with each $0 \le \widehat{F}_{\mathrm{RQMC},m,k}(y) \le 1$ for all y, we have that the overall CDF estimator $\widetilde{F}_{\mathrm{RQMC},m,r}$ in (2.27) at ξ satisfies a CLT

$$\sqrt{r}\left[\widetilde{F}_{\mathrm{RQMC},m,r}(\xi) - p\right] \Rightarrow N(0, \psi^2_{\mathrm{RQMC},m}) \quad \text{as } r \to \infty, \text{ with } m \text{ fixed.} \tag{2.29}$$

By applying the theoretical framework developed in [21], we can establish the following properties of the corresponding quantile estimator $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ in (2.28).

Theorem 1 If $f(\xi) > 0$, then for any fixed $m > 0$,

$$\widetilde{\xi}_{\mathrm{RQMC},m,r} = \xi + \frac{p - \widetilde{F}_{\mathrm{RQMC},m,r}(\xi)}{f(\xi)} + R'_r, \tag{2.30}$$

$$\text{with } \sqrt{r}R'_r \Rightarrow 0, \quad \text{as } r \to \infty. \tag{2.31}$$

Moreover, for each fixed $m > 0$,

$$\sqrt{r}\left[\widetilde{\xi}_{\mathrm{RQMC},m,r} - \xi\right] \Rightarrow N(0, \tau^2_{\mathrm{RQMC},m}) \quad \text{as } r \to \infty, \tag{2.32}$$

where $\tau^2_{\mathrm{RQMC},m} = \psi^2_{\mathrm{RQMC},m}/f^2(\xi)$ for $\psi^2_{\mathrm{RQMC},m}$ in Equation (2.29). If in addition $\{r(\widetilde{\xi}_{\mathrm{RQMC},m,r} - \xi)^2 : r \geq 1\}$ is uniformly integrable (e.g., p. 338 of [17]), then

$$\mathrm{RMSE}\left[\widetilde{\xi}_{\mathrm{RQMC},m,r}\right] = r^{-1/2}\tau^2_{\mathrm{RQMC},m} + o(r^{-1/2}) = O(r^{-1/2}) \tag{2.33}$$

as $r \to \infty$ for fixed m.

Note that (2.30) and (2.31) establish a Bahadur representation for $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ as $r \to \infty$ with m fixed. Also, even though $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ is biased for fixed r and m, the CLT in (2.32) is centered at the true quantile ξ, in contrast to the CLT Equation (2.21). Comparing (2.33) with Equation (2.25), we see the advantage of the RQMC quantile estimator $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ in Equation (2.28) over $\overline{\xi}_{\mathrm{RQMC},m,r}$ in Equation (2.20): as $r \to \infty$ with m fixed, the RMSE of $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ shrinks to 0 but the RMSE of $\overline{\xi}_{\mathrm{RQMC},m,r}$ does not. The RMSE of $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ converges at rate $r^{-1/2}$, which is the standard MC rate. But the numerical results in the next section show that RQMC can lead to substantially smaller MSE than MC, so we view RQMC as an MSE-reduction technique.

2.6 Numerical Results

We now present results from running numerical experiments with the model in Example 2 from Section 2.1, which is motivated by studies of nuclear power plants undergoing hypothesized accidents; e.g., see [28], [118], and [2]. The goal is to estimate the 0.05-quantile ξ of the safety margin $Y \sim F$, where $Y = C - L$. Let G denote the joint CDF of (L,C), and let G_L and G_C be the marginal CDFs of the load L and the capacity C, respectively. As in [28] and [118], we assume that L and C are independent, and we specify G_C as triangular with support $[1800, 2600]$ and mode 2200.

Paper [2] assumes that the load's marginal distribution G_L is a *mixture* of $t = 4$ lognormals, which we also use. Specifically, for each $s = 1, 2, \ldots, t$, let $G_{L,\langle s\rangle}$ be the CDF of $L_{\langle s\rangle} = \exp(\mu_{\langle s\rangle} + \sigma_{\langle s\rangle} Z_{\langle s\rangle})$, where $Z_{\langle s\rangle} \sim N(0,1)$, and $\mu_{\langle s\rangle}$ and $\sigma_{\langle s\rangle} > 0$ are given constants, so $L_{\langle s\rangle}$ has a lognormal distribution. Our experiments set $\mu_{\langle s\rangle} = 7.4 + 0.1s$ and $\sigma_{\langle s\rangle} = 0.01 + 0.01s$, which are as in [96]. Then define G_L as a mixture of $G_{L,\langle s\rangle}$, $1 \le s \le t$; i.e., $G_L(y) = \sum_{s=1}^{t} \lambda_{\langle s\rangle} G_{L,\langle s\rangle}(y)$ for given positive constants $\lambda_{\langle s\rangle}$, $1 \le s \le t$, summing to 1. We set $\lambda_{\langle 1\rangle} = 0.99938 \times 0.9981 \times 0.919$, $\lambda_{\langle 2\rangle} = 0.00062$, $\lambda_{\langle 3\rangle} = 0.99938 \times 0.9981 \times 0.081$, and $\lambda_{\langle 4\rangle} = 0.99938 \times 0.0019$, where the factors in each product match branching probabilities given in an event tree in Figure 2 of [28].

We can define the function w_Y in Equation (2.1) to take $d = 3$ i.i.d. uniform inputs to generate $Y = w_Y(U_1, U_2, U_3)$. The function w_Y uses U_1 and U_2 to generate the load $L \sim G_L$ as follows. First it employs U_1 to generate a discrete random variable K with support $R = \{1, 2, \ldots, t\}$ and probability mass function $P(K = s) = \lambda_{\langle s\rangle}$. If $K = s$, then generate L having CDF $G_{L,\langle s\rangle}$, which is lognormal. Specifically, if $K = s$, let $L = \exp(\mu_{\langle s\rangle} + \sigma_{\langle s\rangle} \Phi^{-1}(U_2))$ where Φ is the $N(0,1)$ CDF. Also, w_Y generates the capacity as $C = G_C^{-1}(U_3)$. Finally, w_Y returns $Y = C - L$. Because of the analytical tractability of the model, we were able to numerically compute the 0.05-quantile as $\xi = 11.79948572$.

To examine the effect of the problem dimension d on RQMC, we also considered another stochastically equivalent version of the model with larger d. Specifically, we artificially increase the dimension by generating the lognormal $L_{\langle s\rangle}$ as the exponential of a sum of $d' = 20$ independent normals with different marginal variances so that for each s, the sum of the d' marginal variances equals $\sigma^2_{\langle s\rangle}$. To specify the different marginal variances, we sampled d' independent chi-square random variables $V_{s,1}, V_{s,2}, \ldots, V_{s,d'}$, and set the marginal variance of the jth summand as $\sigma^2_{\langle s\rangle} V_{s,j} / \sum_{j'=1}^{d'} V_{s,j'}$. The overall problem dimension is then $d = 22$. We used the same marginal variances when running multiple independent replications.

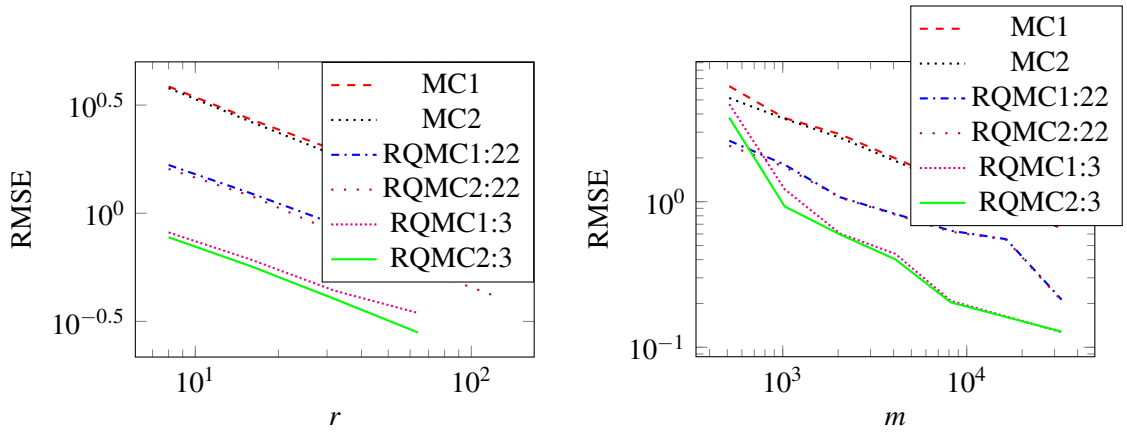


Figure 2.1 The left plot shows RMSE for fixed $m = 4096$ as r increases, and the right graph displays RMSE for fixed $r = 32$ as m increases. Both plots have log-log scale.

Figure 2.1 presents two log-log plots of the RMSE for the estimators of ξ using MC or RQMC, where we estimated the RMSEs from 10^3 independent replications. Each estimator is based on a total of $n = rm$ evaluations of the response function w_Y. For RQMC, m represents the point-set size, and r is the number of randomizations. For each version of the model dimension d (= 3 or 22), we compare two RQMC estimators of ξ, denoted RQMCv:d for v = 1 or 2 in the figure. RQMC1:d is the estimator $\overline{\xi}_{\mathrm{RQMC},m,r}$ in Equation (2.20), and RQMC2:d is the estimator $\widetilde{\xi}_{\mathrm{RQMC},m,r}$ in Equation (2.28). In the following, we often simplify notation by omitting the "$:d$" in the discussions. For RQMC, we used a lattice point set with a random shift modulo 1 for randomization, utilizing the code of [67]. We also ran experiments employing a Sobol' point set with a random digital shift ([124] shows this is a good practical choice), and the results (not shown) are qualitatively similar.

For MC, we also computed two different estimators, denoted MCv for v = 1 or 2 in the figure. (For MC, we plot the results for only $d = 22$ and not for $d = 3$ because the results are stochastically equivalent.) The v = 1 estimator (i.e., MC1) averages r independent p-quantile estimators, where each estimator is calculated by inverting a CDF estimator based on a sample of size m. The v = 2 estimator computes a single p-quantile estimator by inverting a CDF estimator from all $n = rm$ outputs.

The graph on the left side of Figure 2.1 has fixed $m = 4096$ and r increasing, and the plot on the right has fixed $r = 32$ and m increasing. (We chose $r = 32$ for the right plot as this corresponds to the common (but sometimes inadequate) rule of thumb that the asymptotics of a CLT roughly start holding for sample sizes at least 30.) To interpret the left plot of Figure 2.1, recall that MSE decomposes as bias squared plus variance. For RQMC2, RMSE shrinks at rate $O(r^{-1/2})$ as r grows with m fixed by Equation (2.33). But for RQMC1, Equation (2.24) shows that the bias contribution to MSE does not change as r increases with m fixed. (For $\mathrm{v} = 1, 2$, the RMSE of MCv behaves like that of RQMCv.) For small r, the variance dominates the MSE for $\mathrm{v} = 1$, and the RMSEs for $\mathrm{v} = 1$ and 2 are close. For $\mathrm{v} = 1$, as r grows, the variance shrinks, but the bias does not change, so the bias eventually dominates the RMSE, and the RMSEs for $\mathrm{v} = 1$ and 2 then separate for large r. For RQMC, the curves for $d = 3$ and $d = 22$ are qualitatively similar, but the RMSEs for $d = 3$ are smaller, where the plots for $v = 2$ have equal slope but different intercepts.

For the right plot of Figure 2.1, for MC or RQMC, the $\mathrm{v} = 1$ and 2 estimators' RMSEs differ for small m. But as m grows, we see that the RMSE for $\mathrm{v} = 1$ and 2 eventually merge.

CHAPTER 3

QUANTILE ESTIMATION VIA A COMBINATION OF CONDITIONAL MONTE CARLO AND RANDOMIZED QUASI-MONTE CARLO

In this chapter, we consider to estimate quantile by using the combination of *conditional Monte Carlo* (CMC) and *randomized quasi-Monte Carlo* (RQMC). The chapter's remainder is organized as follows. In Section 3.1, we define the mathematical setting and introduce an example that we carry all along the paper. In the five following sections, we recall the basic definitions and properties of MC, CMC, QMC, RQMC, and the CMC+RQMC combination, and we explain how each of them works to estimate a quantile. Section 4.6 reports the results of numerical experiments with our running example.

3.1 Mathematical Framework

Consider a random variable Y (also called the *response*) defined by

$$Y = b_Y(U_1, U_2, \ldots, U_d) = b_Y(\mathbf{U}) \tag{3.1}$$

for a given (deterministic) *response function* $b_Y : (0,1)^d \to \mathbb{R}$ with fixed integer $d \geq 1$, where $U_1, U_2, \ldots, U_d$ are i.i.d. $\mathscr{U}(0,1)$ (i.e., uniform on the interval $(0,1)$) random variables, so $\mathbf{U} = (U_1, U_2, \ldots, U_d) \sim \mathscr{U}(0,1)^d$. The function b_Y can be quite complicated, first transforming $\mathbf{U}$ into a random vector $\mathbf{V}$ having a specified joint distribution, and then using $\mathbf{V}$ in detailed computations to finally output a response Y; e.g., see Example 2 below. Let F be the CDF of Y:

$$F(y) = P(Y \leq y) = P(b_Y(\mathbf{U}) \leq y) = \int_{(0,1)^d} \mathbb{I}(b_Y(\mathbf{u}) \leq y)\, \mathrm{d}\mathbf{u} \qquad \text{for all } y \in \mathbb{R},$$

where $\mathbb{I}$ denotes the indicator function. We assume that we know how to compute Y for any realization of $\mathbf{U}$, but we do not know how to compute F or the quantiles analytically or numerically. Our goal is to estimate the p-quantile $\xi = \xi_p$ for a fixed p. We assume

$f(\xi) = F'(\xi)$ exists and is strictly positive, so that $y = \xi$ is the unique root of $F(y) = p$. To estimate ξ, we will estimate F, replace it by its estimator in the equation $F(y) = p$, and take the root as our quantile estimator.

We illustrate the notation through the following example, motivated by a probabilistic safety assessment (PSA) of a nuclear power plant (NPP); see Section 3.2 of [129].

Example 2 A system experiencing a random load L has a random capacity C to withstand the load, and the system suffers damage when $L > C$. We define the system's *safety margin* as $Y = C - L$, and let F denote its CDF. This situation arises, e.g., in a NPP undergoing a hypothesized accident, where L denotes the peak cladding temperature (PCT), and C is the temperature at which the cladding is damaged; e.g., see [28]. Nuclear engineers model L as a random variable as its value depends on unforeseen aspects of how the event progresses. The capacity C is also considered as random because it involves uncertainties regarding the cladding's material properties.

Nuclear engineers employ detailed computer codes ([49]) to generate an observation of (L,C), which often entails numerically solving systems of differential equations. We express this through two deterministic functions $b_L : (0,1)^d \to \mathbb{R}$ and $b_C : (0,1)^d \to \mathbb{R}$ as

$$(L,C) = (b_L(\mathbf{U}), b_C(\mathbf{U})), \tag{3.2}$$

where $\mathbf{U} \sim \mathscr{U}(0,1)^d$. Note that we evaluate the functions b_L and b_C using the same uniform vector $\mathbf{U}$, so L and C can be dependent. The response function in (3.1) is $b_Y(\mathbf{u}) = b_C(\mathbf{u}) - b_L(\mathbf{u})$ for $\mathbf{u} \in (0,1)^d$, so we obtain an observation of the safety margin as $Y = b_Y(\mathbf{U})$.

Let $\theta = P(Y < 0)$, which is the probability of sustaining damage during an event, and suppose that a regulator has specified that θ must not exceed a given threshold p, e.g., $p = 0.05$. We can restate the requirement that $\theta \leq p$ in terms of a quantile: the p-quantile ξ of Y must satisfy $\xi \geq 0$. A quantile provides information that can be more easily interpretable than the probability θ because ξ is expressed in the same units (temperature) as L and

C. Indeed, nuclear risk studies are often performed using quantiles; e.g., see U.S. Nuclear Regulatory Commission [129]. □

3.2 Monte Carlo

To estimate ξ via MC, we generate n i.i.d. random vectors $\mathbf{U}_i \sim \mathscr{U}(0,1)^d$, $i = 1,2,\ldots,n$, put $Y_i = b_Y(\mathbf{U}_i)$ for each i, and define the empirical CDF by

$$\widehat{F}_{\mathrm{MC},n}(y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(Y_i \le y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(b_Y(\mathbf{U}_i) \le y), \tag{3.3}$$

which is an unbiased MC estimator of $F(y)$. The MC estimator of the p-quantile ξ is then

$$\widehat{\xi}_{\mathrm{MC},n} = \widehat{F}^{-1}_{\mathrm{MC},n}(p). \tag{3.4}$$

It can be easily computed as $\widehat{\xi}_{\mathrm{MC},n} = Y_{\lceil np \rceil:n}$ where $\lceil \cdot \rceil$ denotes the ceiling function and $Y_{1:n} \le Y_{2:n} \le \cdots \le Y_{n:n}$ are the n observations $Y_1, Y_2, \ldots, Y_n$ sorted by increasing order.

Under our assumption that $f(\xi) > 0$, the MC p-quantile estimator obeys a *central limit theorem* (CLT):

$$\sqrt{n}[\widehat{\xi}_{\mathrm{MC},n} - \xi] \Rightarrow N(0, \tau^2_{\mathrm{MC}}) \quad \text{as } n \to \infty,$$

where $\Rightarrow$ denotes convergence in distribution (Chapter 5 of [17]),

$$\tau^2_{\mathrm{MC}} = \frac{\mathrm{Var}[\mathbb{I}(Y \le \xi)]}{f^2(\xi)} = \frac{p(1-p)}{f^2(\xi)} \tag{3.5}$$

is the CLT's *asymptotic variance*, and $\mathrm{Var}[\cdot]$ is the variance operator; see Section 2.3.3 of [117].

We can construct a confidence interval (CI) for ξ as follows. Let $\widehat{\tau}^2_{\mathrm{MC},n}$ be a consistent estimator of τ^2_{MC}, e.g., as in [18]; i.e., $\widehat{\tau}^2_{\mathrm{MC},n} \Rightarrow \tau^2_{\mathrm{MC}}$ as $n \to \infty$. Then for a fixed confidence level $0 < \beta < 1$, we define a CI for ξ as $J_{\mathrm{MC},n} = [\widehat{\xi}_{\mathrm{MC},n} \pm z_\beta \widehat{\tau}_{\mathrm{MC},n}/\sqrt{n}]$, where $z_\beta = \Phi^{-1}(1-(1-\beta)/2)$ and Φ is the $N(0,1)$ CDF. The CI is asymptotically valid in

the sense that $\lim_{n\to\infty} P(\xi \in J_{\mathrm{MC},n}) = \beta$. There are also other approaches for constructing a CI for ξ via MC; see [93].

As shown by [10], the *root mean square error* (RMSE) of $\widehat{\xi}_{\mathrm{MC},n}$, defined as $(E[(\widehat{\xi}_{\mathrm{MC},n} - \xi)^2])^{1/2}$, decreases at the canonical MC rate of $O(n^{-1/2})$ as $n \to \infty$. This is a rather slow rate of convergence, as reducing the error by a factor of 10 requires a 100-fold increase in n.

3.3 Conditional Monte Carlo

When estimating a mean, CMC (Section V.4 of [9]) reduces sampling error by analytically integrating out some of the variability. [95] applies CMC with MC for quantile estimation (also see [8]). [26] combine CMC with Latin hypercube sampling (LHS; [90]), which can be viewed as a variant of stratified sampling in high dimensions, or as a primitive form of RQMC [105].

To apply CMC for quantile estimation, we apply CMC to estimate F, then we invert this CMC estimator of F to obtain the CMC quantile estimator. More specifically, let $\mathscr{G}$ be a sigma-field not containing all of the information needed to compute Y (we discuss specific examples of $\mathscr{G}$ below). Using iterated expectations (p. 448 of [17]), we can write

$$F(y) = E[\mathbb{I}(Y \le y)] = E[E[\mathbb{I}(Y \le y) \mid \mathscr{G}]] = E[q(y,\mathscr{G})], \tag{3.6}$$

where $q(y,\mathscr{G}) := P(Y \le y \mid \mathscr{G})$, which we assume is easily computable. Averaging n i.i.d. realizations of $q(y,\mathscr{G})$, say $q(y,\mathscr{G}_1),\ldots,q(y,\mathscr{G}_n)$, gives the following unbiased CMC estimator of $F(y)$:

$$\widehat{F}_{\mathrm{CMC},n}(y) = \frac{1}{n}\sum_{i=1}^{n} q(y,\mathscr{G}_i). \tag{3.7}$$

By inverting $\widehat{F}_{\mathrm{CMC},n}$, we get the CMC p-quantile estimator

$$\widehat{\xi}_{\mathrm{CMC},n} = \widehat{F}^{-1}_{\mathrm{CMC},n}(p). \tag{3.8}$$

Computing $\widehat{\xi}_{\mathrm{CMC},n}$ typically requires employing a numerical root-finding method, such as the bisection or Newton's method (Chapter 7 of [102]), which incurs extra computation cost.

In our setting where $Y = b_Y(\mathbf{U})$ with $\mathbf{U} \sim \mathscr{U}(0,1)^d$, we may for example define $\mathscr{G}$ as (the sigma-field generated by) a random vector

$$\mathbf{Z} \equiv (Z_1, Z_2, \ldots, Z_l) = b_{\mathbf{Z}}(\mathbf{U}) \tag{3.9}$$

for a deterministic function $b_{\mathbf{Z}} : (0,1)^d \to \mathbb{R}^l$ for some $l \geq 1$, using the *same* $\mathscr{U}(0,1)^d$ vector $\mathbf{U}$ from (3.1), making Y and $\mathbf{Z}$ generally dependent. Thus, $b_{\mathbf{Z}}(\mathbf{U})$ has all of the details needed to produce one realization of the information contained in the sigma-field $\mathscr{G}$. Although the function $b_{\mathbf{Z}}$ is defined to take as input all d coordinates from $\mathbf{U}$ in (3.1), it may not use all of them. By writing $\mathbf{Z}_i = b_{\mathbf{Z}}(\mathbf{U}_i)$, (3.7) becomes

$$\widehat{F}_{\mathrm{CMC},n}(y) = \frac{1}{n}\sum_{i=1}^{n} q(y, \mathbf{Z}_i) = \frac{1}{n}\sum_{i=1}^{n} q(y, b_{\mathbf{Z}}(\mathbf{U}_i)). \tag{3.10}$$

Example 2 (continued) To apply CMC to our example, we define the conditioning vector in (3.9) as $\mathbf{Z} = L$, so $b_{\mathbf{Z}}(\mathbf{U}) = b_L(\mathbf{U})$, where b_L is from (3.2). As in [28], we now assume that the load L and the capacity C are independent. From a modeling perspective, it is reasonable to have L and C independent as L is determined by random variables that affect how the hypothesized accident unfolds, whereas C depends on uncertainties about the material properties of the cladding. Let G_C be the marginal CDF of C. We then write the CDF F of the safety margin $Y = C - L$ as

$$F(y) = P(C - L \leq y) = E[P(C \leq y + L \mid L)] = E[G_C(y + L)] \tag{3.11}$$

by the independence of L and C. Thus, we have that $q(y, \mathbf{Z}) = G_C(y + L) = G_C(y + b_L(\mathbf{U}))$, and (3.10) gives $\widehat{F}_{\mathrm{CMC},n}(y) = (1/n)\sum_{i=1}^{n} G_C(y + b_L(\mathbf{U}_i))$ as the CMC estimator of the CDF.

The independence of L and C holds, e.g., when the two functions $b_L(\mathbf{u})$ and $b_C(\mathbf{u})$ in (3.2) depend only on disjoint subsets of the components of $\mathbf{u} = (u_1, u_2, \ldots, u_d) \in (0,1)^d$. Specifically, consider a subset $\mathscr{S} \subset \{1,2,\ldots,d\}$ of the coordinate indices, and let $\bar{\mathscr{S}} = \{1,2,\ldots,d\} - \mathscr{S}$ be its complement. For a vector $\mathbf{u} \in (0,1)^d$, let $\mathbf{u}_{\mathscr{S}} \in (0,1)^{|\mathscr{S}|}$ (resp., $\mathbf{u}_{\bar{\mathscr{S}}} \in (0,1)^{|\bar{\mathscr{S}}|}$) be the projection of $\mathbf{u}$ onto the coordinates in $\mathscr{S}$ (resp., $\bar{\mathscr{S}}$). Now assume that there exist functions $b_L^* : (0,1)^{|\mathscr{S}|} \to \mathbb{R}$ and $b_C^* : (0,1)^{|\bar{\mathscr{S}}|} \to \mathbb{R}$ such that $b_L(\mathbf{u}) = b_L^*(\mathbf{u}_{\mathscr{S}})$ and $b_C(\mathbf{u}) = b_C^*(\mathbf{u}_{\bar{\mathscr{S}}})$ for every $\mathbf{u} \in (0,1)^d$. Then $L = b_L(\mathbf{U}) = b_L^*(\mathbf{U}_{\mathscr{S}})$ and $C = b_C(\mathbf{U}) = b_C^*(\mathbf{U}_{\bar{\mathscr{S}}})$ are independent because $\mathbf{U}_{\mathscr{S}}$ and $\mathbf{U}_{\bar{\mathscr{S}}}$ are, as $\mathbf{U} \sim \mathscr{U}(0,1)^d$ has independent components. □

As noted by [95], when $f(\xi) > 0$, the CMC p-quantile estimator obeys the CLT $\sqrt{n}[\widehat{\xi}_{\mathrm{CMC},n} - \xi] \Rightarrow N(0, \tau^2_{\mathrm{CMC}})$ as $n \to \infty$, where

$$\tau^2_{\mathrm{CMC}} = \mathrm{Var}[q(\xi, \mathscr{G})]/f^2(\xi). \tag{3.12}$$

A variance decomposition (e.g., p. 456 of [17]) implies that for each $y \in \mathbb{R}$,

$$\begin{aligned}\mathrm{Var}[\mathbb{I}(Y \le y)] &= \mathrm{Var}[E[\mathbb{I}(Y \le y) \mid \mathscr{G}]] + E[\mathrm{Var}[\mathbb{I}(Y \le y) \mid \mathscr{G}]] \\ &= \mathrm{Var}[q(y, \mathscr{G})] + E[\mathrm{Var}[\mathbb{I}(Y \le y) \mid \mathscr{G}]] \;\ge\; \mathrm{Var}[q(y, \mathscr{G})].\end{aligned}$$

As a consequence, $\mathrm{Var}[\widehat{F}_{\mathrm{CMC},n}(y)] = \mathrm{Var}[q(y,\mathscr{G})]/n \le \mathrm{Var}[\mathbb{I}(Y \le y)]/n = \mathrm{Var}[\widehat{F}_{\mathrm{MC},n}(y)]$ for each $y \in \mathbb{R}$, and by taking $y = \xi$, this also gives $\tau^2_{\mathrm{CMC}} \le \tau^2_{\mathrm{MC}}$ by (3.5) and (3.12). This shows that CMC reduces the asymptotic variance of the p-quantile estimator compared to MC, but the RMSE of $\widehat{\xi}_{\mathrm{CMC},n}$ still decreases at the canonical MC rate of $O(n^{-1/2})$, although with a smaller hidden constant.

3.4 Quasi-monte Carlo

QMC replaces the n independent random points $\mathbf{U}_i$ by a set $\mathscr{P}_n$ of n *deterministic* points $\mathbf{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,d}) \in [0,1)^d$ which cover the unit hypercube more evenly than typical independent random points, in the sense that their empirical distribution has a *low*

discrepancy with respect to the uniform distribution over $[0,1)^d$, lower than for typical independent random points. Here we take the interval $[0,1)$ closed on the left because most constructions have points with coordinates equal to 0, but this will disappear when the points are randomized with RQMC. (In practice, we want to avoid 0 and 1 because it causes a problem when generating random variates by inversion from distributions having an infinite tail.) The most common constructions of low-discrepancy point sets are *integration lattices* [120] and *digital nets* [99, 25].

QMC theory was developed mainly for when we want to approximate an integral of the form $\gamma := \int_{[0,1)^d} h(\mathbf{u})\,\mathrm{d}\mathbf{u}$ for some function $h : [0,1)^d \to \mathbb{R}$ by the average $\widehat{\gamma}_{\mathrm{QMC},n} = (1/n)\sum_{i=1}^{n} h(\mathbf{u}_i)$ over the points $\mathbf{u}_i \in \mathscr{P}_n$. There is a large variety of Cauchy-Schwarz-type inequalities of the form

$$\left|\widehat{\gamma}_{\mathrm{QMC},n} - \gamma\right| \le D(\mathscr{P}_n) \cdot V(h),$$

where $D(\mathscr{P}_n)$ measures the discrepancy of $\mathscr{P}_n$ and $V(h)$ measures the variation of h. One special case is the classical *Koksma-Hlawka inequality* in which $D(\mathscr{P}_n)$ is the star discrepancy $D^*(\mathscr{P}_n)$ and V is the Hardy-Krause variation V_{HK} [99, Section 2.2], but it is impractical because these quantities are too hard to compute. However, other more easily computable discrepancies, together with matching definitions of V, also exist [25, 70, 75]. Explicit point-set constructions can achieve $D(\mathscr{P}_n) = O(n^{-\alpha+\varepsilon})$ for any $\varepsilon > 0$, often for $\alpha = 1$ (e.g., for the star discrepancy) and for $\alpha > 1$ for some discrepancies. Then, for any h for which $V(h) < \infty$ for the corresponding V, the integration error converges to 0 at the same rate (at worst) as $D(\mathscr{P}_n)$. It is true that the hidden constant in the convergence order can increase very quickly with the dimension d, but there are nevertheless many large-dimensional integrands for which QMC is much more accurate than MC. Typically, this occurs when h can be decomposed approximately as a sum of low-dimensional functions; see [110, 70, 25] and [75].

Reference [109] approximate F and ξ as in (3.3) and (3.4), with the random $\mathbf{U}_i$'s replaced by the QMC points $\mathbf{u}_i$. This gives

$$\widehat{F}_{\mathrm{QMC},n}(y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(b_Y(\mathbf{u}_i) \le y) \tag{3.13}$$

and $\widehat{\xi}_{\mathrm{QMC},n} = \widehat{F}^{-1}_{\mathrm{QMC},n}(p)$. One could think of applying QMC theory to show that $\widehat{F}_{\mathrm{QMC},n}(y)$ converges to $F(y)$ faster than $\widehat{F}_{\mathrm{MC},n}(y)$. But this CDF estimator corresponds to applying QMC to the integrand $h(\mathbf{u}) = h_y(\mathbf{u}) := \mathbb{I}(b_Y(\mathbf{u}) \le y)$, which is a discontinuous function of $\mathbf{u}$. This discontinuity implies that the variation $V(h_y)$ is typically infinite, so there is no error bound, and the gain from QMC is usually very small in that situation; see, e.g., Sections 15.12 and 16.5 of [108] and [71].

We saw earlier that CMC reduces the variance of a CDF estimator, but much more importantly, it also provides a very powerful opportunity to make $q(y, b_{\mathbf{Z}}(\mathbf{u}))$ continuous and smooth in $\mathbf{u}$. Then, its combination with QMC could give a much better approximation of F than the QMC approximation in (3.13), and also a much better quantile estimator. Assuming that the conditioning sigma-field $\mathscr{G}$ in (3.6) is defined by $\mathbf{Z} = b_{\mathbf{Z}}(\mathbf{U})$ in (3.9), the idea is simply to replace each random $\mathbf{U}_i$ in (3.10) with $\mathbf{u}_i \in \mathscr{P}_n$ to get a CDF approximation $\widehat{F}_{\mathrm{CQ},n}(y)$, whose inversion gives an approximation $\widehat{\xi}_{\mathrm{CQ},n} = \widehat{F}^{-1}_{\mathrm{CQ},n}(p)$ of the p-quantile. (Here, the subscript "CQ" is an abbreviation for the combination CMC+QMC.)

CMC has already been applied in the setting of estimating the derivative of an expectation or of a quantile with respect to some model parameter [77, 34, 33]. It was also merged with QMC or RQMC to estimate a mean ([73] and [47]). [78] obtain large gains by combining CMC with RQMC for density estimation.

A major limitation of QMC, however, is that it does not provide an easily computable bound or estimate for the error on the CDF and for the errors $|\widehat{\xi}_{\mathrm{QMC},n} - \xi|$ and $|\widehat{\xi}_{\mathrm{CQ},n} - \xi|$ on the quantile approximations. This motivates RQMC, described next.

3.5 Randomized Quasi-monte Carlo

RQMC turns QMC into a variance reduction method by randomizing the points in a way that they retain their low discrepancy as a group, while each individual point has the uniform distribution over the unit hypercube [104, 106, 73, 74, 79, 71]. By doing $r \geq 2$ independent randomizations, one can obtain an unbiased estimator for the variance of the CDF estimator by using the sample variance across randomizations, and eventually an estimator for the variance of the quantile estimator and a confidence interval for the quantile. But as noted by [61], one has to be careful in how to adapt the RQMC approach for estimating a mean to estimating the quantile ξ instead, because the quantile estimator is generally biased.

We now provide more details on RQMC. To have a fair comparison with the MC and CMC estimators in Equations (3.4) and (3.8), which are based on n evaluations of an integrand, we want to implement RQMC with the same total number of integrand evaluations. Thus, we start with a low-discrepancy point set of size $m < n$, which we randomize $r = n/m$ times, where we assume that $r \geq 2$ is integer-valued. Specifically, let $\mathscr{P}_m = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\} \subset [0,1)^d$ be a low-discrepancy point set of size m. We randomize this point set $r = n/m$ independent times to obtain r independent randomized point sets $\mathscr{P}_m^{(k)}$, $k = 1, 2, \ldots, r$, with $\mathscr{P}_m^{(k)} = \{\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_m^{(k)}\}$ and each $\mathbf{X}_i^{(k)} = (X_{i,1}^{(k)}, X_{i,2}^{(k)}, \ldots X_{i,d}^{(k)})$ has the uniform distribution over $(0,1)^d$.

Appropriate randomizations of QMC point sets, which preserve the low-discrepancy properties, depend on the type of point-set construction. For integration lattices, a random shift modulo 1 is the most common and appropriate approach [23, 124, 73]. It works as follows. For the kth randomization, we generate a single random vector $\mathbf{S}_k = (S_{k,1}, S_{k,2}, \ldots, S_{k,d}) \sim \mathscr{U}(0,1)^d$ and we add it to all the points $\mathbf{u}_i$ of the original set $\mathscr{P}_m$, modulo 1, component-wise. This gives a randomized point set $\mathscr{P}_m^{(k)} = \{\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_m^{(k)}\}$, where each $\mathbf{X}_i^{(k)} = ((u_{i,1} + S_{k,1}) \bmod 1, (u_{i,2} + S_{k,2}) \bmod 1, \ldots, (u_{i,d} + S_{k,d}) \bmod 1)$. Clearly, each $\mathbf{X}_i^{(k)}$ has the uniform distribution over $(0,1)^d$, and the point set

keeps its lattice structure; it becomes a shifted lattice. We get independent randomizations simply by taking r independent random shifts $\mathbf{S}_k$, $k = 1, 2, \ldots, r$. When estimating a mean, the resulting estimator is called a *randomly-shifted lattice rule*, and its variance properties have been studied extensively [23, 124, 73, 75].

For digital nets, the uniformity is usually measured in terms of equidistribution of the points for certain families of rectangular boxes, and a random shift does not preserve these properties. Other types of randomizations such as *random digital shifts* [80, 25, 71], and various scrambles [104, 106, 108], do preserve them.

To construct an asymptotically valid CI based on RQMC, it is desirable for the RQMC estimator to obey a CLT with a Gaussian limit. When estimating a mean, [87] proves such a CLT as the size m of the point set grows large for RQMC using a digital net with full nested scrambling [104]. However, this scrambling is computationally more expensive than random shifts (digital or modulo 1).

For mean estimators based on randomly-shifted lattice rules, [76] determine that the limiting distribution (as $m \to \infty$) is generally not Gaussian, with histograms from numerical studies exhibiting asymmetry and multiple modes. Also, to specify the number r of randomizations, practitioners often choose r as not too large, e.g., $r = 10$ or 20 at most, to be able take a larger m to exploit the power of RQMC. But this can result in a poor variance estimator and a CI with poor coverage, especially when considering biased estimators (as for a quantile) or when studying tail behavior (e.g., a p-quantile with $p \approx 0$ or $p \approx 1$). Thus, as the computation budget increases, to obtain an RQMC CI that is asymptotically valid, it can be useful to have the number r of randomizations increase to infinity to ensure a CLT with a Gaussian limit and to allow for consistent estimation of the CLT's asymptotic variance.

We now describe one possible approach, which we call RQMC1, for quantile estimation based on the typical RQMC method for estimating a mean, as described at the

beginning of this section. For each randomization $k = 1, 2, \ldots, r$, compute

$$\widehat{F}_{\mathrm{RQMC1},m,k}(y) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(b_Y(\mathbf{X}_i^{(k)}) \le y) \tag{3.14}$$

as an unbiased CDF estimator from the randomized point set $\mathscr{P}_m^{(k)}$. Inverting $\widehat{F}_{\mathrm{RQMC1},m,k}$ yields $\widehat{\xi}_{\mathrm{RQMC1},m,k} = \widehat{F}^{-1}_{\mathrm{RQMC1},m,k}(p)$, and the $\widehat{\xi}_{\mathrm{RQMC1},m,k}$, $k = 1, 2, \ldots, r$, are i.i.d. Then the *RQMC1 quantile estimator* is

$$\overline{\xi}_{\mathrm{RQMC1},m,r} = \frac{1}{r}\sum_{k=1}^{r} \widehat{\xi}_{\mathrm{RQMC1},m,k}. \tag{3.15}$$

Even if (3.14) is unbiased for $F(y)$, the nonlinearity of the inverse operation leads to $\widehat{\xi}_{\mathrm{RQMC1},m,k}$ having bias in general, so under the assumption that $E[|\widehat{\xi}_{\mathrm{RQMC1},m,k}|] < \infty$, the law of large numbers implies that

$$\overline{\xi}_{\mathrm{RQMC1},m,r} \overset{\text{w.p.1}}{\to} E[\widehat{\xi}_{\mathrm{RQMC1},m,k}] \neq \xi, \quad \text{as } r \to \infty \text{ with } m \text{ fixed;} \tag{3.16}$$

i.e., $\overline{\xi}_{\mathrm{RQMC1},m,r}$ *converges to the wrong value*. While the bias of $\overline{\xi}_{\mathrm{RQMC1},m,r}$ decreases to zero as m grows, (3.16) considers the asymptotic regime in which r increases with m fixed, causing the problems. If $\tau'^2_{\mathrm{RQMC1},m} \equiv \mathrm{Var}[\widehat{\xi}_{\mathrm{RQMC1},m,k}] < \infty$, then $\overline{\xi}_{\mathrm{RQMC1},m,r}$ obeys a CLT $\sqrt{r}(\overline{\xi}_{\mathrm{RQMC1},m,r} - E[\widehat{\xi}_{\mathrm{RQMC1},m,k}]) \Rightarrow \mathscr{N}(0, \tau'^2_{\mathrm{RQMC1},m})$ as $r \to \infty$ with m fixed, where $\tau'^2_{\mathrm{RQMC1},m} = \mathrm{Var}[\widehat{\xi}_{\mathrm{RQMC1},m,k}]$ depends on m. However, the CLT uses $E[\widehat{\xi}_{\mathrm{RQMC1},m,k}]$ as the centering constant rather than the true quantile ξ, so a CI based on the CLT will have poor coverage, as the CI is anchored at $\overline{\xi}_{\mathrm{RQMC1},m,r}$, which on average does not equal the true value ξ.

To address this, [61] propose a second RQMC quantile estimator that *does converge* to ξ in the asymptotic setting of (3.16). Rather than use each randomization to compute a quantile estimator, as in (3.15), we instead employ all r randomizations to compute a *single overall* CDF estimator, which is inverted to obtain a single overall quantile estimator.

Specifically, first define the CDF estimator

$$\widetilde{F}_{\mathrm{RQMC2},m,r}(y) = \frac{1}{r}\sum_{k=1}^{r} \widehat{F}_{\mathrm{RQMC1},m,k}(y) = \frac{1}{rm}\sum_{k=1}^{r}\sum_{i=1}^{m} \mathbb{I}(b_Y(\mathbf{X}_i^{(k)}) \le y)$$

based on all rm evaluations of b_Y. Inverting this leads to the *RQMC2 quantile estimator*

$$\widetilde{\xi}_{\mathrm{RQMC2},m,r} = \widetilde{F}^{-1}_{\mathrm{RQMC2},m,r}(p). \tag{3.17}$$

As with the RQMC1 quantile estimator, the RQMC2 estimator $\widetilde{\xi}_{\mathrm{RQMC2},m,r}$ is also biased. But unlike the bias of $\overline{\xi}_{\mathrm{RQMC1},m,r}$, which does not shrink as $r \to \infty$ with m fixed, leading to the convergence to the wrong value in Equation (3.16), the bias of $\widetilde{\xi}_{\mathrm{RQMC2},m,r}$ decreases to 0. Indeed, it does so fast enough to ensure that under our assumption that $f(\xi) > 0$, the RQMC2 quantile estimator obeys a CLT

$$\sqrt{r}[\widetilde{\xi}_{\mathrm{RQMC2},m,r} - \xi] \Rightarrow \mathcal{N}(0, \tau^2_{\mathrm{RQMC2},m}), \quad \text{as } r \to \infty \text{ with } m \text{ fixed}, \tag{3.18}$$

as noted in [61], and the asymptotic variance is $\tau^2_{\mathrm{RQMC2},m} = \mathrm{Var}[\widehat{F}_{\mathrm{RQMC1},m,k}(\xi)]/f^2(\xi)$, which depends on m. Thus, the CLT (3.18) has the desired centering constant ξ, in contrast to the CLT for the RQMC1 estimator $\overline{\xi}_{\mathrm{RQMC1},m,r}$.

3.6 Combining CMC With RQMC For Quantile Estimation

We can combine CMC with RQMC for quantile estimation as follows. As in Section 3.4, we assume that the conditioning sigma-field $\mathscr{G}$ in Equation (3.6) is defined by $\mathbf{Z} = b_{\mathbf{Z}}(\mathbf{U})$ in Equation (3.9). Then just as we went from the QMC CDF estimator in Equation (3.13) to the CQ CDF estimator by substituting $\mathbb{I}(b_Y(\mathbf{u}_i) \le y)$ with $q(y, b_{\mathbf{Z}}(\mathbf{u}_i))$, we similarly replace $\mathbb{I}(b_Y(\mathbf{X}_i^{(k)}) \le y)$ in Equation (3.14) for the RQMC1 CDF estimator with $q(y, b_{\mathbf{Z}}(\mathbf{X}_i^{(k)}))$ to obtain the combined CMC+RQMC1 (abbreviated CR1) estimator of the CDF from randomization k as

$$\widehat{F}_{\mathrm{CR1},m,k}(y) = \frac{1}{m}\sum_{i=1}^{m} q(y, b_{\mathbf{Z}}(\mathbf{X}_i^{(k)})).$$

Inverting $\widehat{F}_{\mathrm{CR1},m,k}$ yields $\widehat{\xi}_{\mathrm{CR1},m,k} = \widehat{F}^{-1}_{\mathrm{CR1},m,k}(p)$, where $\widehat{\xi}_{\mathrm{CR1},m,k}$, $k = 1,2,\ldots,r$, are i.i.d. Then our *CR1 quantile estimator* is

$$\overline{\xi}_{\mathrm{CR1},m,r} = \frac{1}{r}\sum_{k=1}^{r} \widehat{\xi}_{\mathrm{CR1},m,k}. \tag{3.19}$$

But the CR1 quantile estimator also suffers from the problems in Equation (3.16) that the RQMC1 quantile estimator has caused by bias (but perhaps less so, as we will see in the numerical results of Section 4.6).

For the combined CMC+RQMC2 (shortened to CR2) quantile estimator, we first compute the CR2 CDF estimator by

$$\widetilde{F}_{\mathrm{CR2},m,r}(y) = \frac{1}{r}\sum_{k=1}^{r} \widehat{F}_{\mathrm{CR1},m,k}(y) = \frac{1}{rm}\sum_{k=1}^{r}\sum_{i=1}^{m} q(y, b_{\mathbf{Z}}(\mathbf{X}_i^{(k)}))$$

based on all *rm* evaluations of function q. Inverting this leads to our *CR2 quantile estimator*

$$\widetilde{\xi}_{\mathrm{CR2},m,r} = \widetilde{F}^{-1}_{\mathrm{CR2},m,r}(p). \tag{3.20}$$

Using the framework in [21], we can prove that under our assumption that $f(\xi) > 0$, the CR2 quantile estimator obeys a CLT

$$\sqrt{r}[\widetilde{\xi}_{\mathrm{CR2},m,r} - \xi] \Rightarrow \mathcal{N}(0, \tau^2_{\mathrm{CR2},m}), \quad \text{as } r \to \infty \text{ with } m \text{ fixed}, \tag{3.21}$$

with asymptotic variance $\tau^2_{\mathrm{CR2},m} = \mathrm{Var}[\widehat{F}_{\mathrm{CR1},m,k}(\xi)]/f^2(\xi)$, which depends on m.

3.7 Numerical Results

We now present numerical results from a stylized version of the model in Example 2, where the goal is to estimate the 0.05-quantile ξ of the system's safety margin $Y = C - L$. We first describe the model, which is also considered in [96], [2], and [61]. As in some actual NPP PSA studies (e.g., [28]), the CDF G_C of the capacity C is assumed triangular with support $[1800, 2600]$ and mode 2200, with L and C independent. The marginal CDF G_L of the load

L is a mixture of $t = 4$ lognormals, as in [96]. Specifically, for each $s = 1, 2, \ldots, t$, let $G_{L,\langle s\rangle}$ be the CDF of $L_{\langle s\rangle} = \exp(\mu_{\langle s\rangle} + \sigma_{\langle s\rangle} Z_{\langle s\rangle})$, where $Z_{\langle s\rangle} \sim \mathcal{N}(0,1)$, and $\mu_{\langle s\rangle}$ and $\sigma_{\langle s\rangle} > 0$ are given constants, so $L_{\langle s\rangle}$ has a lognormal distribution. Our experiments set $\mu_{\langle s\rangle} = 7.4 + 0.1s$ and $\sigma_{\langle s\rangle} = 0.01 + 0.01s$. Then G_L is defined as $G_L(y) = \sum_{s=1}^{t} \lambda_{\langle s\rangle} G_{L,\langle s\rangle}(y)$ for given positive constants $\lambda_{\langle s\rangle}$, $1 \le s \le t$, summing to 1. We set $\lambda_{\langle 1\rangle} = 0.99938 \times 0.9981 \times 0.919$, $\lambda_{\langle 2\rangle} = 0.00062$, $\lambda_{\langle 3\rangle} = 0.99938 \times 0.9981 \times 0.081$, and $\lambda_{\langle 4\rangle} = 0.99938 \times 0.0019$, where the factors in each product match branching probabilities given in an event tree used in an NPP PSA study by [28].

For this model, we can define the response function b_Y in Equation (3.1) to generate the safety margin $Y = b_Y(U_1, U_2, U_3)$ using just $d = 3$ i.i.d. uniform inputs: one to choose the component in the load mixture G_L, another to generate the appropriate lognormal, and the third to sample the capacity C. For CMC, we condition on L, as in Equation (3.11), so Equation (3.10) uses $q(y, \mathbf{Z}) = G_C(y + L) = G_C(y + b_L(\mathbf{U}))$, as in Example 2 of Section 3.3. Thus, CMC requires only 2 uniforms, as the one for sampling C is integrated out by G_C. The analytical tractability of the model allows us to numerically compute the 0.05-quantile as $\xi = 11.79948572$.

Given that the effectiveness of RQMC typically degrades as the problem dimension d increases (Section 15.9 of [108]), we also consider for comparison a stochastically equivalent artificial model with larger d, defined as follows. We generate the lognormal $L_{\langle s\rangle}$ by exponentiating the sum of 20 independent normals, with parameters chosen so that the sum still has mean $\mu_{\langle s\rangle}$ and variance $\sigma^2_{\langle s\rangle}$. Thus, b_Y in Equation (3.1) now has $d = 22$.

We consider quantile estimation using two versions ($v = 1$ and 2) each of RQMC and CMC+RQMC. RQMCv denotes version v of RQMC, so the RQMC1 estimator is $\overline{\xi}_{\mathrm{RQMC1},m,r}$ in Equation (3.15), and the RQMC2 estimator is $\widetilde{\xi}_{\mathrm{RQMC2},m,r}$ in Equation (3.17). We have two versions of the CMC+RQMC quantile estimator: the CR1 estimator $\overline{\xi}_{\mathrm{CR1},m,r}$ in Equation (3.19), and the CR2 estimator $\widetilde{\xi}_{\mathrm{CR2},m,r}$ in Equation (3.20). In each case, our RQMC point set is a randomly-shifted lattice rule of size m, and we make r independent

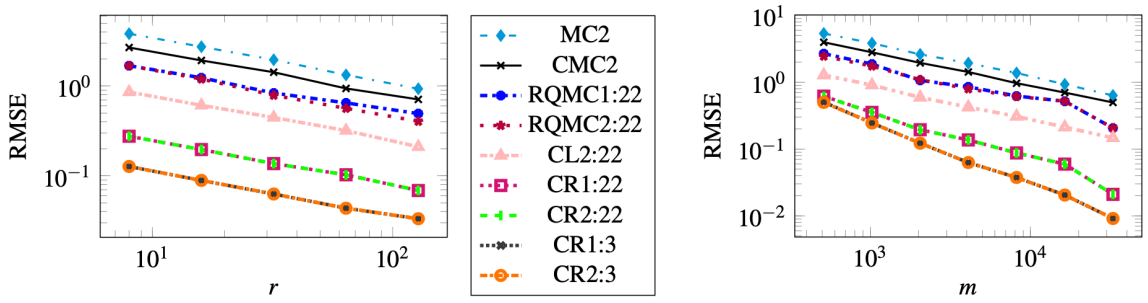


Figure 3.1 The left panel shows the RMSE for fixed $m = 4096$ as r increases, and the right panel displays the RMSE for fixed $r = 32$ as m increases. Both plots have log-log scale. The notation "$:d$" in the legend specifies the problem dimension without applying CMC. For CRv:d with $d = 3$ and 22, the plots for $v = 1$ and 2 lie on top of each other.

random shifts. For MC and CMC, we consider only a single version $v = 2$, in which we compute a single CDF estimator based on all $n = rm$ outputs, and invert this to get the MC2 and CMC2 quantile estimators. We also include CMC+LHS ([26]) corresponding to version $v = 2$, which we write as CL2. (We also ran experiments with CL1, but omit those results as they are indistinguishable from CL2 in plots with our r and m values.)

Figure 3.1 presents two sets of log-log plots of the RMSE, estimated from 10^3 independent replications, of the various quantile estimators. In the left plots, the number r of randomizations increases, with a point set of fixed size $m = 4096$. On the right side, m grows with fixed $r = 32$. The figures show the RQMC results for different problem dimensions, where d ($= 3$ or $= 22$) in the notation "$:d$" represents the problem dimension without applying CMC. For MC and CMC, changing d does not affect the RMSE, so Figure 3.1 presents results for only $d = 22$.

We now compare the mean-square error (MSE) of each $v = 2$ method for $r = 64$ and $m = 4096$. With MC2 as the baseline, CMC2 (resp., RQMC2:22, CL2:22, and CR2:22) reduces MSE by a factor of 2.0 (resp., 5.6, 17.3, and 168). Thus, while RQMC2 improves on MC2 and CMC2, the combination of CR2 performs substantially better, illustrating the benefits of a smoother integrand for RQMC. Moreover, RQMC2:3 (not shown in Figure 3.1) and CR2:3 reduce MSE (compared to MC2) by factors of 21.5

and 928, respectively, thus demonstrating the impact of the problem dimension d on RQMC's effectiveness. The MSE of CR2:22 is about a tenth of that of CL2:22, showing CMC+RQMC's advantage over CMC+LHS.

As previously explained in [61], RQMC1 has issues as r increases with m fixed, as seen in Equation (3.16), which we next explain. To understand the left plots in Figure 3.1, recall that the MSE decomposes as the sum of bias squared and variance. The variance of the RQMC1 quantile estimator shrinks to 0 as $r \to \infty$ with fixed m. But because quantile estimators are generally biased, the RQMC1 bias does not decrease because m is fixed. Thus, the RMSE of RQMC1 will converge to a *strictly positive value* as $r \to \infty$ with m fixed. We can see this start to happen in the left plots in Figure 3.1, where the RMSE of the RQMC1 estimator is leveling off as r increases. (The other $v = 1$ estimators also eventually suffer from the same problem, although it may not be clear for the range of r considered.) In contrast, the RQMC2 quantile estimator obeys a CLT (as $r \to \infty$ with m fixed) with centering constant ξ, so its RMSE shrinks to 0. The left plots of Figure 3.1 demonstrate the steady decrease in RMSE of the $v = 2$ estimators.

The right panel of Figure 3.1 shows that as m increases, the rate (i.e., slope) at which the RMSE decreases for RQMC (especially for CRv:3) can be better than for MCv. This demonstrates that RQMC can not only reduce variance but also improve convergence rates. However, CMC+LHS, although having lower RMSE than MC and CMC, still converges at the standard MC rate [71].

CHAPTER 4

MONTE CARLO METHODS FOR ECONOMIC CAPITAL

In this chapter, we analyze large-sample properties of EC estimators obtained via SRS only, IS only, MSIS, IS using a defensive mixture, and a double estimator using both SRS and IS to estimate both the quantile and the mean, establishing Bahadur-type representations for the EC estimators and proving they obey central limit theorems. We also provide asymptotic theory comparing the estimators when the loss is the sum of a large number of independent and identically distributed random variables. Numerical results, including for a large portfolio credit risk model with dependent obligors, complement the theory.

The rest of the chapter unfolds as follows. Section 4.1 gives our mathematical framework. Section 4.2 presents the SRS estimator of η. It further establishes a type of large-sample [12] representation for the estimator, and proves a CLT, which we also do for the other methods considered. Section 4.3 applies IS to estimate η. Section 4.4 describes the methods that combine IS and SRS: MSIS (Section 4.4.1), ISDM (Section 4.4.2), and DE (Section 4.4.3). We provide in Section 4.5 our theoretical asymptotic comparisons of the estimators of η, ξ, and μ when Y is the sum of m i.i.d. random variables as $m \to \infty$ with the quantile level p simultaneously approaching 1. Section 4.6 gives numerical/simulation results demonstrating the benefits of MSIS over the other methods, with Section 4.6.1 considering the model from Section 4.5, and Section 4.6.2 examining a more complicated model, an extension of the PCRM from [40]. Chapter 6 gives concluding remarks. Proofs are collected in appendices, which also provides additional numerical results and describes the simulation methodology used on the PCRM in Section 4.6.2. Our theorems on the Bahadur representations and CLTs for the SRS, IS, and MSIS estimators appear without proofs in [60], which also describes batching and sectioning methods [9, Section V.5], briefly covered here, to construct large-sample confidence intervals (CIs) for η. [60] do not consider any of the material in Sections 4.4.2, 4.4.3, 4.5, 4.6 and the appendices.

4.1 Mathematical Framework

Let Y be a random variable for the loss of a credit-portfolio model over a given time horizon, and let F be its *cumulative distribution function* (CDF). Assume that F is unknown or computationally intractable, but we have a simulation model that generates an observation of $Y \sim F$, where $\sim$ denotes "is distributed as". Let $\mu = E[Y]$ be the mean of $Y \sim F$. For a CDF H and $0 < q < 1$, we define the *q-quantile* of H as $H^{-1}(q) = \inf\{y : H(y) \geq q\}$; e.g., the median μ' is the 0.5-quantile, also known as the 50th percentile. Our goal is to use simulation to estimate the EC $\eta = \xi - \mu$, where $\xi = F^{-1}(p)$ for a given $0 < p < 1$. (Note that $\xi \equiv \xi_p$ and $\eta \equiv \eta_p$ depend on p, but we omit the subscript p to simplify notation.)

Sometimes but not always, we assume that the loss Y has the form

$$Y = c(\mathbf{X}) \tag{4.1}$$

for a known function $c : \mathbb{R}^d \to \mathbb{R}$ with $d \geq 1$, and random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ having a specified joint CDF G, where G can allow the components of $\mathbf{X}$ to be dependent and non-identically distributed. We view the function c in (4.1) as a (complicated) computer code, transforming an input $\mathbf{X} \sim G$ into a loss $Y \sim F$.

For example, Section 4.6.2 will consider a multi-factor portfolio-credit-risk model as in [40], [15], and [88], in which the loss Y has a form in (4.1), with mutually independent components in $\mathbf{X}$, defined as follows. There are $m \geq 1$ obligors, and dependence among the default events across obligors is induced through common factors. Let $\mathbf{Z} = (Z_1, \ldots, Z_r)$ be a column vector of $r \geq 1$ *systematic risk factors*, which are i.i.d. $N(0,1)$ random variables, modeling global, country, and sector factors that impact all obligors, where $N(q, s^2)$ represents a normal random variable with mean q and variance s^2. For each $k = 1, 2, \ldots, m$, let ε_k be another independent $N(0,1)$ random variable denoting the *idiosyncratic risk* associated with obligor k. The *loading factors* are specified constant row vectors $\mathbf{a}_k = (a_{k,j} : j = 1, 2, \ldots, r)$, $k = 1, 2, \ldots, m$, satisfying $\mathbf{a}_k \mathbf{a}_k^\top \leq 1$ for each k, where $\top$ denotes transpose. Let $b_k = (1 - \mathbf{a}_k \mathbf{a}_k^\top)^{1/2}$, so $\mathbf{a}_k \mathbf{Z} + b_k \varepsilon_k \sim N(0,1)$ for each k. Let $S > 0$ be

another independent random variable denoting a *common shock* affecting all obligors. For each $k = 1, 2, \ldots, m$, obligor k defaults if and only if $(\mathbf{a}_k \mathbf{Z} + b_k \varepsilon_k)/S > w_k$ for a constant w_k chosen so that obligor k has a specified marginal default probability p_k. [40] and [15] assume that the *loss given default* (LGD) of obligor k is a constant c_k, but they state their methods also allow LGD to be stochastic (under certain conditions), as we will have in Section 4.6.2. For obligor k, let J_k be another independent random variable, and define the LGD for obligor k as $v_k(\mathbf{Z}, S, \varepsilon_1, \ldots, \varepsilon_m, J_k)$ for a given function $v_k : \mathbb{R}^{r+m+2} \to \mathbb{R}_+$. Therefore, the LGD may depend on J_k, as well as the systematic and idiosyncratic risk factors and common shock, as in [5] and [30]. Finally, let $\mathbf{X} = (\mathbf{Z}, S, \varepsilon_1, \ldots, \varepsilon_m, J_1, \ldots, J_m)$, which has $d = r + 2m + 1$ independent components, and the function c in Equation (4.1) for the total loss is

$$c(\mathbf{X}) = \sum_{k=1}^{m} v_k(\mathbf{Z}, S, \varepsilon_1, \ldots, \varepsilon_m, J_k)\, I\left(\frac{\mathbf{a}_k \mathbf{Z} + b_k \varepsilon_k}{S} > w_k\right), \tag{4.2}$$

where $I(\cdot)$ is the indicator function, which equals 1 (resp., 0) if its argument is true (resp., false).

4.2 Simple Random Sampling

We begin with the application of SRS to estimate η, and in this section, the results do not require that the loss $Y \sim F$ has the form in Equation (4.1). Let $Y_1, Y_2, \ldots, Y_n$ be a *random sample* of size n from F; i.e., $Y_1, Y_2, \ldots, Y_n$ are i.i.d. with CDF F. When Y has the form in Equation (4.1), we generate $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ as i.i.d. copies of $\mathbf{X} \sim G$, and let $Y_i = c(\mathbf{X}_i)$ for each $i = 1, 2, \ldots, n$. In general, define the SRS estimator of the mean as the sample mean

$$\widehat{\mu}_{\mathrm{SRS},n} = \frac{1}{n} \sum_{i=1}^{n} Y_i. \tag{4.3}$$

We define the SRS p-quantile estimator $\widehat{\xi}_{\mathrm{SRS},n}$ by inverting the *empirical CDF* $\widehat{F}_{\mathrm{SRS},n}$:

$$\widehat{\xi}_{\mathrm{SRS},n} = \widehat{F}^{-1}_{\mathrm{SRS},n}(p), \quad \text{where } \widehat{F}_{\mathrm{SRS},n}(y) = \frac{1}{n} \sum_{i=1}^{n} I(Y_i \le y). \tag{4.4}$$

Then the SRS estimator of the EC $\eta = \xi - \mu$ is

$$\widehat{\eta}_{\mathrm{SRS},n} = \widehat{\xi}_{\mathrm{SRS},n} - \widehat{\mu}_{\mathrm{SRS},n}. \tag{4.5}$$

We can compute $\widehat{\xi}_{\mathrm{SRS},n}$ by order statistics. Let $Y_{(1)} \le Y_{(2)} \le \cdots \le Y_{(n)}$ be the sorted values of $Y_1, Y_2, \ldots, Y_n$, and then $\widehat{\xi}_{\mathrm{SRS},n} = Y_{(\lceil np \rceil)}$, where $\lceil \cdot \rceil$ is the ceiling (i.e., round-up) function. For simplicity, we do not consider other SRS quantile estimators, e.g., by interpolating $\widehat{F}_{\mathrm{SRS},n}$ ([10]).

While the estimator $\widehat{\mu}_{\mathrm{SRS},n}$ in Equation (4.3) of the mean is a sample average, the p-quantile estimator $\widehat{\xi}_{\mathrm{SRS},n} = \widehat{F}^{-1}_{\mathrm{SRS},n}(p)$ is *not*, so the EC estimator $\widehat{\eta}_{\mathrm{SRS},n}$ in (4.5) is also not a sample average, complicating its analysis. However, [12] shows that $\widehat{\xi}_{\mathrm{SRS},n}$ can be well approximated by a sample average of i.i.d. quantities when the sample size n is large, and we will do the same for $\widehat{\eta}_{\mathrm{SRS},n}$. To accomplish this, define f as the derivative (when it exists) of the CDF F. Also let $\Rightarrow$ represent convergence in distribution (e.g., Chapter 5 of [17]). Then (see Section 2.5 of [117]) if $f(\xi) > 0$, the p-quantile estimator satisfies

$$\widehat{\xi}_{\mathrm{SRS},n} = \xi - \frac{1}{f(\xi)} \left[\widehat{F}_{\mathrm{SRS},n}(\xi) - p \right] + R_n, \tag{4.6}$$

$$\text{with } \sqrt{n} R_n \Rightarrow 0 \quad \text{as } n \to \infty. \tag{4.7}$$

If F is twice differentiable at ξ, then [62] proves that for either choice of sign below,

$$\limsup_{n \to \infty} \pm \frac{n^{3/4} R_n}{(\log \log n)^{3/4}} = \frac{2^{5/4} [p(1-p)]^{1/4}}{3^{3/4} f(\xi)} \quad \text{with probability 1.} \tag{4.8}$$

Note that (4.8) implies (4.7), and we call (4.6) with (4.7) (resp., (4.8)) a *weak* (resp., *strong*) *Bahadur representation* for $\widehat{\xi}_{\mathrm{SRS},n}$. The key point of (4.6)–(4.8) is that they permit analyzing the large-sample properties of $\widehat{\xi}_{\mathrm{SRS},n}$ through the simpler $\widehat{F}_{\mathrm{SRS},n}(\xi)$, which is a sample average of i.i.d. terms by (4.4). As next seen, the SRS EC estimator $\widehat{\eta}_{\mathrm{SRS},n}$ has similar *Bahadur-type representations* and obeys a CLT; see Appendix B for the proof.

Theorem 2 Suppose that $Y_1, Y_2, \ldots$ are i.i.d. with CDF F, and F is differentiable at ξ with $f(\xi) > 0$.

(i) The SRS EC estimator in (4.5) then satisfies

$$\widehat{\eta}_{\mathrm{SRS},n} = \eta - \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{f(\xi)}[I(Y_i \le \xi) - p] + [Y_i - \mu]\right) + R_n \tag{4.9}$$

with R_n from Equation (4.6), so Equation (4.7) holds. If also F is twice differentiable at ξ, then Equation (4.8) further holds.

(ii) If in addition $\sigma^2_{\mathrm{SRS}} \equiv \mathrm{Var}[Y] < \infty$, then

$$\sqrt{n}\left[\widehat{\eta}_{\mathrm{SRS},n} - \eta\right] \Rightarrow N(0, \zeta^2_{\mathrm{SRS}}) \quad \text{as } n \to \infty, \text{ where } \zeta^2_{\mathrm{SRS}} = \frac{\chi^2_{\mathrm{SRS}}}{f^2(\xi)} + \sigma^2_{\mathrm{SRS}} + 2\frac{\gamma_{\mathrm{SRS}}}{f(\xi)}, \tag{4.10}$$

with $\chi^2_{\mathrm{SRS}} = p(1-p)$, and $\gamma_{\mathrm{SRS}} = \mathrm{Cov}[I(Y \le \xi), Y] = E[I(Y \le \xi)Y] - p\mu$.

The Bahadur-type representations in Theorem 2(i) give useful insight into the large-sample behavior of $\widehat{\eta}_{\mathrm{SRS},n}$, showing that approximating $\widehat{\eta}_{\mathrm{SRS},n} - \eta$ by a sample mean of the i.i.d. terms results in a remainder R_n that vanishes faster than $1/\sqrt{n}$ (by Equation (4.7) or Equation (4.8)). This then implies the CLT in (4.10) when $\sigma^2_{\mathrm{SRS}} < \infty$.

Under an additional assumption that F has a density f and the second derivative of F is bounded in a neighborhood of ξ, [85] prove that the SRS estimators of a quantile and the mean obey a joint CLT, and [32] also shows the same under the weaker additional assumption that the density f is continuous at ξ. While (4.10) follows from either result, the Bahadur-type representations in Theorem 2(i) can be used to show the asymptotic validity of a sectioning CI for η, as in [60].

4.3 Importance Sampling

When $p \approx 1$, estimators of $\xi = F^{-1}(p)$ and the corresponding

EC $\eta = \xi - \mu$ may have large variance, motivating the use of a variance-reduction technique. We consider applying IS, but other VRTs are also possible. To use IS and the methods in the next section, we assume Y has the form in Equation (4.1) from now on.

The mean of Y is then $\mu = E_G[c(\mathbf{X})]$, where, for any CDF $G^\dagger$ on $\mathbb{R}^d$, $E_{G^\dagger}$ (resp., $\mathrm{Var}_{G^\dagger}$, $\mathrm{Cov}_{G^\dagger}$) is the expectation (resp., variance, covariance) operator when $\mathbf{X} \sim G^\dagger$. Let $\widetilde{G}$ be a CDF on $\mathbb{R}^d$ such that (the measure of) G is absolutely continuous [17, p. 422] with respect to $\widetilde{G}$. By a *change of measure*,

$$\mu = E_G[c(\mathbf{X})] = \int_{\mathbb{R}^d} c(\mathbf{x}) \mathrm{d}G(\mathbf{x}) = \int_{\mathbb{R}^d} c(\mathbf{x}) \frac{\mathrm{d}G(\mathbf{x})}{\mathrm{d}\widetilde{G}(\mathbf{x})} \mathrm{d}\widetilde{G}(\mathbf{x}) = E_{\widetilde{G}}[c(\mathbf{X})L(\mathbf{X})], \text{ for } L(\mathbf{x}) = \frac{\mathrm{d}G(\mathbf{x})}{\mathrm{d}\widetilde{G}(\mathbf{x})} \tag{4.11}$$

as the *likelihood ratio* (LR), for $\mathbf{x} \in \mathbb{R}^d$. To estimate μ via IS, we sample i.i.d. $\mathbf{X}_i \sim \widetilde{G}$, $i = 1, 2, \ldots, n$, and

$$\widehat{\mu}_{\mathrm{IS},n} = \frac{1}{n} \sum_{i=1}^{n} c(\mathbf{X}_i) L(\mathbf{X}_i) \tag{4.12}$$

is an unbiased estimator of μ by (4.11). (IS reduces to SRS when $\widetilde{G} = G$ as then $L(\mathbf{x}) \equiv 1$.)

Suppose that under both G and $\widetilde{G}$, the components of $\mathbf{X} = (X_1, \ldots, X_d)$ are mutually independent. Then for G_j (resp., $\widetilde{G}_j$) denoting the marginal CDF of X_j under G (resp., $\widetilde{G}$), we have $G(\mathbf{x}) = \prod_{j=1}^d G_j(x_j)$ and $\widetilde{G}(\mathbf{x}) = \prod_{j=1}^d \widetilde{G}_j(x_j)$ for $\mathbf{x} = (x_1, \ldots, x_d)$. If we further suppose that each G_j (resp., $\widetilde{G}_j$) has a density or probability mass function g_j (resp., $\tilde{g}_j$), the likelihood ratio in (4.11) becomes $L(\mathbf{x}) = \prod_{j=1}^d \frac{g_j(x_j)}{\tilde{g}_j(x_j)}$.

To estimate the p-quantile ξ by IS, we use an approach of [41]: first apply IS to estimate the CDF F, and then invert the estimated CDF to obtain the IS quantile estimator. Specifically, write

$$1 - F(y) = E_G[I(c(\mathbf{X}) > y)] = E_{\widetilde{G}}[I(c(\mathbf{X}) > y) L(\mathbf{X})] \tag{4.13}$$

through a change of measure. By (4.13), we obtain an unbiased estimator of $F(y)$ as $\widehat{F}_{\mathrm{IS},n}(y)$, with

$$\widehat{F}_{\mathrm{IS},n}(y) = 1 - \frac{1}{n} \sum_{i=1}^{n} I(c(\mathbf{X}_i) > y) L(\mathbf{X}_i), \quad \text{and} \quad \widehat{\xi}_{\mathrm{IS},n} = \widehat{F}^{-1}_{\mathrm{IS},n}(p), \tag{4.14}$$

where $\mathbf{X}_i \sim \widetilde{G}$, $i = 1,2,\ldots,n$, are the same as in Equation (4.12). We call $\widehat{F}_{\mathrm{IS},n}(y)$ and $\widehat{\xi}_{\mathrm{IS},n}$ the *IS estimators* of $F(y)$ and ξ, respectively. To compute $\widehat{\xi}_{\mathrm{IS},n}$, let $Y_i = c(\mathbf{X}_i)$, and let $Y_{1:n} \le Y_{2:n} \le \cdots \le Y_{n:n}$ be the sorted values of $Y_1, Y_2, \ldots, Y_n$. Defining $\mathbf{X}_{i::n}$ as the $\mathbf{X}_j$ corresponding to $Y_{i:n}$, we then have $\widehat{\xi}_{\mathrm{IS},n} = Y_{i_p:n}$, with i_p the greatest integer for which $\sum_{\ell=i_p}^{n} L(\mathbf{X}_{\ell::n}) \ge n(1-p)$. Reference [21] establish that the quantile estimator obtained via a combination of IS and stratified sampling obeys a weak Bahadur representation, with $\widehat{\xi}_{\mathrm{IS},n}$ in Equation (4.14) being a special case of IS only; i.e., their Theorem 4.2 shows that if

$$\text{there exist constants } \varepsilon > 0 \text{ and } \lambda > 0 \text{ such that } E_{\widetilde{G}}[I(c(\mathbf{X}) > \xi - \lambda)L^{2+\varepsilon}(\mathbf{X})] < \infty, \tag{4.15}$$

then

$$\widehat{\xi}_{\mathrm{IS},n} = \xi - \frac{1}{f(\xi)}[\widehat{F}_{\mathrm{IS},n}(\xi) - p] + \tilde{R}_n, \quad \text{with} \quad \sqrt{n}\tilde{R}_n \Rightarrow 0 \quad \text{as } n \to \infty. \tag{4.16}$$

The fact that $F(y) = E_G[I(c(\mathbf{X}) \le y)] = E_{\widetilde{G}}[I(c(\mathbf{X}) \le y)L(\mathbf{X})]$ suggests another CDF estimator, $\widehat{F}'_{\mathrm{IS},n}(y) = \frac{1}{n}\sum_{i=1}^{n} I(c(\mathbf{X}_i) \le y)L(\mathbf{X}_i)$, with each $\mathbf{X}_i \sim \widetilde{G}$, which leads to another p-quantile estimator $\widehat{\xi}'_{\mathrm{IS},n} = \widehat{F}'^{-1}_{\mathrm{IS},n}(p)$. Theorem 4.1 of [21] (resp., [122]) establishes a weak (resp., strong) Bahadur representation for $\widehat{\xi}'_{\mathrm{IS},n}$. When estimating the p-quantile with $p \approx 1$ using IS, [41] shows that for a simple example, the p-quantile estimator $\widehat{\xi}_{\mathrm{IS},n}$ in Equation (4.14) has smaller asymptotic variance than the estimator $\widehat{\xi}'_{\mathrm{IS},n}$. (But $\widehat{\xi}'_{\mathrm{IS},n}$ can have smaller asymptotic variance than $\widehat{\xi}_{\mathrm{IS},n}$ when $p \approx 0$.)

The IS estimator of the EC is then

$$\widehat{\eta}_{\mathrm{IS},n} = \widehat{\xi}_{\mathrm{IS},n} - \widehat{\mu}_{\mathrm{IS},n}, \tag{4.17}$$

with both $\widehat{\xi}_{\mathrm{IS},n}$ and $\widehat{\mu}_{\mathrm{IS},n}$ computed from the same i.i.d. sample $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$, with each $\mathbf{X}_i \sim \widetilde{G}$. The following result, proven in Appendix C, shows that $\widehat{\eta}_{\mathrm{IS},n}$ has a Bahadur-type representation and obeys a CLT.

Theorem 3 Suppose that $Y \sim F$ has the form in Equation (4.1), and $f(\xi) > 0$. Suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ are i.i.d. with CDF $\widetilde{G}$, where (the measure induced by) G is absolutely continuous with respect to $\widetilde{G}$. Also suppose that Equation (4.15) holds for $L(\mathbf{x})$ in Equations (4.11) and (4.13). Then the following hold.

(i) The IS EC estimator in Equation (4.17) satisfies

$$\widehat{\eta}_{\mathrm{IS},n} = \eta - \frac{1}{n}\sum_{i=1}^{n}\left[\frac{(1 - I(c(\mathbf{X}_i) > \xi)L(\mathbf{X}_i)) - p}{f(\xi)} + c(\mathbf{X}_i)L(\mathbf{X}_i) - \mu\right] + \tilde{R}_n$$
$$\text{with } \sqrt{n}\tilde{R}_n \Rightarrow 0 \text{ as } n \to \infty. \tag{4.18}$$

(ii) If in addition $\sigma^2_{\mathrm{IS}} \equiv \mathrm{Var}_{\widetilde{G}}[c(\mathbf{X})L(\mathbf{X})] < \infty$, then $\sqrt{n}\left[\widehat{\eta}_{\mathrm{IS},n} - \eta\right] \Rightarrow N(0, \zeta^2_{\mathrm{IS}})$ as $n \to \infty$, where

$$\zeta^2_{\mathrm{IS}} = \frac{\chi^2_{\mathrm{IS}}}{f^2(\xi)} + \sigma^2_{\mathrm{IS}} - 2\frac{\gamma_{\mathrm{IS}}}{f(\xi)}, \text{ with } \chi^2_{\mathrm{IS}} \equiv \mathrm{Var}_{\widetilde{G}}[I(c(\mathbf{X}) > \xi)L^2(\mathbf{X})], \text{ and} \tag{4.19}$$

$$\gamma_{\mathrm{IS}} \equiv \mathrm{Cov}_{\widetilde{G}}[I(c(\mathbf{X}) > \xi)L(\mathbf{X}), c(\mathbf{X})L(\mathbf{X})] = E_G[I(c(\mathbf{X}) > \xi)c(\mathbf{X})L(\mathbf{X})] - (1-p)\mu. \tag{4.20}$$

4.4 Methods that Combine SRS and IS

Section 4.3 estimates ξ and μ from the same data generated from IS distribution $\widetilde{G}$, but the resulting estimator of $\eta = \xi - \mu$ can have large variance. When $p \approx 1$, ξ is a property of the right tail of F, whereas μ measures the distribution's central tendency. Thus, a VRT designed to analyze only the tail of F may fare poorly in estimating its mean, and vice versa. A single IS method may not estimate both ξ and μ well.

When $p \approx 1$, the heuristic reason that an IS CDF $\widetilde{G}$ for $\mathbf{X}$ designed to estimate only ξ can do poorly for μ arises from the LR in Equation (4.11) often being immense. To see why, first express the second moment of the IS estimator of μ as $m_2 \equiv E_{\widetilde{G}}[c^2(\mathbf{X})L^2(\mathbf{X})] = E_G[c^2(\mathbf{X})L(\mathbf{X})]$ by a change of measure. The original CDF G assigns much of its probability to points $\mathbf{x}$ with $c(\mathbf{x})$ near the mean μ. But $\widetilde{G}$ shifts most of that mass to

values $\mathbf{x}'$ with $c(\mathbf{x}') \approx \xi$, making points $\mathbf{x}$ with $c(\mathbf{x}) \approx \mu$ rare under $\widetilde{G}$. Thus, the LR $L(\mathbf{x}) = \mathrm{d}G(\mathbf{x})/\mathrm{d}\widetilde{G}(\mathbf{x})$ is enormous for these common $\mathbf{x}$ under G, leading to $m_2 = E_G[c^2(\mathbf{X})L(\mathbf{X})]$ and the variance $\sigma^2_{\mathrm{IS}} = \mathrm{Var}_{\widetilde{G}}[c(\mathbf{X})L(\mathbf{X})]$ of the IS estimator of μ being large (the first moment is unchanged by Equation (4.11)).

4.4.1 Measure-Specific Importance Sampling

To address these issues, Measure-Specific Importance Sampling (MSIS) estimates only ξ by IS and independently estimates μ using SRS. [42] apply MSIS to estimate a ratio of means, in which only one corresponds to a rare event and is thus handled via IS, and the other (non-rare) mean is simulated independently without IS. More generally, we can use one VRT to estimate ξ and another to estimate μ, where we may employ VRTs other than IS.

We next give the details of MSIS. For an overall sample size n, we specify a fraction $0 < \delta < 1$ of the sample size to estimate ξ by IS, and we use SRS to estimate μ with the rest of the sample size. Let δn be the sample size estimating ξ via IS, and $(1-\delta)n$ be the sample size estimating μ by SRS, both which we assume are integer-valued (if not, replace δn and $(1-\delta)n$ by $\lfloor \delta n \rfloor$ and $\lfloor (1-\delta)n \rfloor$, respectively, where $\lfloor \cdot \rfloor$ is the floor function). Let $\widehat{F}_{\mathrm{IS},\delta n}$ be the IS CDF estimator in Equation (4.14) but with sample size δn instead of n, and $\widehat{\xi}_{\mathrm{IS},\delta n} = \widehat{F}^{-1}_{\mathrm{IS},\delta n}(p)$ is the resulting p-quantile estimator. Also let $\widehat{\mu}_{\mathrm{SRS},(1-\delta)n}$ be the SRS estimator of μ in Equation (4.3) with sample size $(1-\delta)n$ instead of n. Then the MSIS estimator of η is

$$\widehat{\eta}_{\mathrm{MSIS},n} = \widehat{\xi}_{\mathrm{IS},\delta n} - \widehat{\mu}_{\mathrm{SRS},(1-\delta)n}. \tag{4.21}$$

We next give a weak Bahadur-type representation and CLT for $\widehat{\eta}_{\mathrm{MSIS},n}$; Appendix D provides the proof.

Theorem 4 Suppose that $Y \sim F$ has the form in Equation (4.1), $f(\xi) > 0$, Equation (4.15) holds, and (the measure induced by) G is absolutely continuous with respect to $\widetilde{G}$. Then the following hold for any fixed $0 < \delta < 1$.

(i) As $n \to \infty$, the MSIS EC estimator in Equation (4.21) satisfies

$$\widehat{\eta}_{\mathrm{MSIS},n} = \eta - \frac{1}{f(\xi)}[\widehat{F}_{\mathrm{IS},\delta n}(\xi) - p] - (\widehat{\mu}_{\mathrm{SRS},(1-\delta)n} - \mu) + \tilde{R}_{n,\delta}, \text{ with } \sqrt{n}\tilde{R}_{n,\delta} \Rightarrow 0. \tag{4.22}$$

(ii) If in addition $\sigma^2_{\mathrm{SRS}} < \infty$, then for χ^2_{IS} from Equation (4.19),

$$\sqrt{n}\left[\widehat{\eta}_{\mathrm{MSIS},n} - \eta\right] \Rightarrow N(0, \zeta^2_{\mathrm{MSIS}}) \text{ as } n \to \infty, \quad \text{where} \quad \zeta^2_{\mathrm{MSIS}} = \frac{\chi^2_{\mathrm{IS}}}{\delta f^2(\xi)} + \frac{\sigma^2_{\mathrm{SRS}}}{1-\delta}. \tag{4.23}$$

In contrast to Equations (4.10) and (4.19), Equation (4.23) has no covariance term as MSIS estimates ξ and μ independently. Also, the value of ζ^2_{MSIS} depends on δ, with $\delta^* = [\chi_{\mathrm{IS}}/f(\xi)]/[\sigma_{\mathrm{SRS}} + (\chi_{\mathrm{IS}}/f(\xi))]$ minimizing ζ^2_{MSIS}. But we do not know σ^2_{SRS}, χ^2_{IS} and $f(\xi)$. However we can use a two-stage procedure, with a pilot run to roughly estimate the unknown parameters, which are used in the second stage with estimated δ^*.

4.4.2 Importance Sampling with a Defensive Mixture Distribution

To estimate simultaneously multiple metrics (including the mean and a tail probability), [50] develops Importance Sampling with a Defensive Mixture Distribution (ISDM), which applies IS, as in Section 4.3, with $\mathbf{X} \sim \widetilde{G}_{\mathrm{ISDM}} \equiv \delta G^* + (1-\delta)G$, where G^* (resp., G) is a new (resp., original) joint CDF for $\mathbf{X}$, and $0 \le \delta \le 1$ is a user-specified constant. We can sample $\mathbf{X}$ from the mixture $\widetilde{G}_{\mathrm{ISDM}}$ by generating $\mathbf{X}$ from G^* (resp., G) with probability δ (resp., $1-\delta$). The ISDM EC estimator has the form (4.17) based on Equations (4.12) and

(4.14), with the LR in Equations (4.11) and (4.13) as

$$L_{\text{ISDM}}(\mathbf{x}) = \frac{\mathrm{d}G(\mathbf{x})}{\mathrm{d}\widetilde{G}_{\text{ISDM}}(\mathbf{x})} = \frac{\mathrm{d}G(\mathbf{x})}{\delta \mathrm{d}G^*(\mathbf{x}) + (1-\delta)\mathrm{d}G(\mathbf{x})}, \text{ so } L_{\text{ISDM}}(\mathbf{x}) \leq \frac{1}{1-\delta} \text{ for all } \mathbf{x}. \tag{4.24}$$

Thus, for $\delta \in (0,1)$, ISDM prevents the LR from being too big (Section 4.4), making $\widetilde{G}_{\text{ISDM}}$ a defensive mixture. A special case of IS, the ISDM EC estimator obeys Theorem 3 ($\delta \in (0,1)$ ensures the assumed absolute continuity and Equation (4.15) hold). When $\delta = 0$ (resp., $\delta = 1$), ISDM reduces to SRS (resp., IS with $\mathbf{X} \sim G^*$).

The CDF G^* itself can be a mixture of r CDFs, so then $\widetilde{G}_{\text{ISDM}}$ mixes $r+1$ CDFs. Other works using a mixture for IS include [103] and [39].

4.4.3 Double Estimator

A *double estimator* (DE) provides another way of combining IS and SRS to estimate the EC. As with MSIS, DE generates an IS (resp., SRS) sample of size δn (resp., $(1-\delta)n$), with the two samples independent. But in contrast to MSIS, DE employs both the IS and SRS samples to estimate both ξ and μ. More specifically, we use the IS sample of size δn to construct estimators $\widehat{\xi}_{\text{IS},\delta n}$ and $\widehat{\mu}_{\text{IS},\delta n}$ in Equations (4.14) and (4.12), respectively. Also, we get estimators $\widehat{\xi}_{\text{SRS},(1-\delta)n}$ and $\widehat{\mu}_{\text{SRS},(1-\delta)n}$ in Equations (4.4) and (4.3), respectively, from the SRS sample of size $(1-\delta)n$. For user-specified constants $\upsilon_1, \upsilon_2 \in [0,1]$, we define the DE EC estimator as

$$\widehat{\eta}_{\text{DE},n} = \left[\upsilon_1 \widehat{\xi}_{\text{IS},\delta n} + \upsilon_1' \widehat{\xi}_{\text{SRS},(1-\delta)n}\right] - \left[\upsilon_2 \widehat{\mu}_{\text{IS},\delta n} + \upsilon_2' \widehat{\mu}_{\text{SRS},(1-\delta)n}\right] \equiv \widehat{\xi}_{\text{DE},n} - \widehat{\mu}_{\text{DE},n}, \tag{4.25}$$

where $\upsilon_1' = 1-\upsilon_1$ and $\upsilon_2' = 1-\upsilon_2$. When $\upsilon_1 = \upsilon_2$, $\widehat{\eta}_{\text{DE},n}$ is a weighted sum of two independent EC estimators: $\widehat{\eta}_{\text{IS},\delta n}$ of Equation (4.17) with an IS sample of size δn and weight $\upsilon_1 = \upsilon_2$, and $\widehat{\eta}_{\text{SRS},(1-\delta)n}$ of Equation (4.5) with an SRS sample of size $(1-\delta)n$ and weight $\upsilon_1' = \upsilon_2'$. But Equation (4.25) also allows $\upsilon_1 \neq \upsilon_2$, and DE becomes MSIS

when $\upsilon_1 = 1$ and $\upsilon_2 = 0$. Also, DE reduces to SRS (resp., IS) when $\upsilon_1 = \upsilon_2 = \delta = 0$ (resp., $\upsilon_1 = \upsilon_2 = \delta = 1$). The following, whose proof appears in Appendix E, gives a Bahadur-type representation and CLT for $\widehat{\eta}_{\mathrm{DE},n}$.

Theorem 5 Suppose that $Y \sim F$ has the form in Equation (4.1) and $f(\xi) > 0$. Suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{(1-\delta)n}$ are i.i.d. with CDF G, and $\mathbf{X}'_1, \mathbf{X}'_2, \ldots, \mathbf{X}'_{\delta n}$ are i.i.d. with CDF $\widetilde{G}$ (and independent of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{(1-\delta)n}$), where (the measure induced by) G is absolutely continuous with respect to $\widetilde{G}$, which satisfies Equation (4.15). Then the following hold for any $\delta, \upsilon_1, \upsilon_2 \in [0,1]$.

(i) The DE estimator $\widehat{\eta}_{\mathrm{DE},n}$ in Equation (4.25), constructed from $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{(1-\delta)n}$ and $\mathbf{X}'_1, \mathbf{X}'_2, \ldots, \mathbf{X}'_{\delta n}$, satisfies

$$\begin{aligned}\widehat{\eta}_{\mathrm{DE},n} = \eta - \Bigg[&\left(\frac{\upsilon_1}{\delta n} \sum_{i=1}^{\delta n} \frac{[1 - I(c(\mathbf{X}'_i) > \xi) L(\mathbf{X}'_i)] - p}{f(\xi)} \right) + \\ &\left(\frac{\upsilon'_1}{(1-\delta)n} \sum_{i=1}^{(1-\delta)n} \frac{I(c(\mathbf{X}_i) \le \xi) - p}{f(\xi)} \right) - \left(\frac{\upsilon_2}{\delta n} \sum_{i=1}^{\delta n} \left[c(\mathbf{X}'_i) L(\mathbf{X}'_i) - \mu \right] \right) \\ &+ \left(\frac{\upsilon'_2}{(1-\delta)n} \sum_{i=1}^{(1-\delta)n} [c(\mathbf{X}_i) - \mu] \right) \Bigg] + (\upsilon_1 \tilde{R}_{\delta n} + \upsilon'_1 R_{(1-\delta)n}), \\ &\text{with } \sqrt{n}(\upsilon_1 \tilde{R}_{\delta n} + \upsilon'_1 R_{(1-\delta)n}) \Rightarrow 0 \quad \text{as } n \to \infty, \end{aligned} \tag{4.26}$$

where $\tilde{R}_{\delta n}$ is from Equation (4.16) and $R_{(1-\delta)n}$ from Equation (4.6).

(ii) If in addition $\sigma^2_{\mathrm{IS}} < \infty$ and $\sigma^2_{\mathrm{SRS}} < \infty$, then $\sqrt{n}[\widehat{\eta}_{\mathrm{DE},n} - \eta] \Rightarrow N(0, \zeta^2_{\mathrm{DE}})$ as $n \to \infty$, where

$$\begin{aligned}\zeta^2_{\mathrm{DE}} = &\left[\frac{\upsilon_1^2}{\delta} \frac{\chi^2_{\mathrm{IS}}}{f^2(\xi)} + \frac{\upsilon_1'^2}{1-\delta} \frac{\chi^2_{\mathrm{SRS}}}{f^2(\xi)} \right] + \left[\frac{\upsilon_2^2}{\delta} \sigma^2_{\mathrm{IS}} + \frac{\upsilon_2'^2}{1-\delta} \sigma^2_{\mathrm{SRS}} \right] + \\ &2 \left[-\frac{\upsilon_1 \upsilon_2}{\delta} \frac{\gamma_{\mathrm{IS}}}{f(\xi)} + \frac{\upsilon'_1 \upsilon'_2}{1-\delta} \frac{\gamma_{\mathrm{SRS}}}{f(\xi)} \right]. \end{aligned} \tag{4.27}$$

For a fixed $\delta \in (0,1)$, the optimal choice of (υ_1, υ_2) to minimize ζ^2_{DE} satisfies

$$(\upsilon_1^*, \upsilon_2^*) = \left(\frac{a_1}{a_0}, \frac{a_2}{a_0}\right), \quad \text{where} \tag{4.28}$$

$$a_0 = \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{V}^{(\mu)}_{\mathrm{IS}} - \mathrm{C}^2_{\mathrm{IS}} + 2\mathrm{C}_{\mathrm{IS}}\mathrm{C}_{\mathrm{SRS}} - \mathrm{C}^2_{\mathrm{SRS}} + \mathrm{V}^{(\xi)}_{\mathrm{IS}}\mathrm{V}^{(\mu)}_{\mathrm{IS}} + \mathrm{V}^{(\xi)}_{\mathrm{IS}}\mathrm{V}^{(\mu)}_{\mathrm{SRS}} + \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{V}^{(\mu)}_{\mathrm{SRS}},$$

$$a_1 = \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{V}^{(\mu)}_{\mathrm{IS}} + \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{V}^{(\mu)}_{\mathrm{SRS}} + \mathrm{V}^{(\mu)}_{\mathrm{IS}}\mathrm{C}_{\mathrm{SRS}} + \mathrm{V}^{(\mu)}_{\mathrm{SRS}}\mathrm{C}_{\mathrm{IS}} + \mathrm{C}_{\mathrm{IS}}\mathrm{C}_{\mathrm{SRS}} - \mathrm{C}^2_{\mathrm{SRS}}, \quad \text{and}$$

$$a_2 = \mathrm{V}^{(\xi)}_{\mathrm{IS}}\mathrm{V}^{(\mu)}_{\mathrm{SRS}} + \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{V}^{(\mu)}_{\mathrm{SRS}} + \mathrm{V}^{(\xi)}_{\mathrm{IS}}\mathrm{C}_{\mathrm{SRS}} + \mathrm{V}^{(\xi)}_{\mathrm{SRS}}\mathrm{C}_{\mathrm{IS}} + \mathrm{C}_{\mathrm{IS}}\mathrm{C}_{\mathrm{SRS}} - \mathrm{C}^2_{\mathrm{SRS}},$$

with $\mathrm{V}^{(\xi)}_{\mathrm{IS}} = \frac{\chi^2_{\mathrm{IS}}}{\delta f^2(\xi)}$, $\mathrm{V}^{(\xi)}_{\mathrm{SRS}} = \frac{\chi^2_{\mathrm{SRS}}}{(1-\delta) f^2(\xi)}$, $\mathrm{V}^{(\mu)}_{\mathrm{IS}} = \frac{\sigma^2_{\mathrm{IS}}}{\delta}$, $\mathrm{V}^{(\mu)}_{\mathrm{SRS}} = \frac{\sigma^2_{\mathrm{SRS}}}{1-\delta}$, $\mathrm{C}_{\mathrm{IS}} = \frac{\gamma_{\mathrm{IS}}}{\delta f(\xi)}$, and $\mathrm{C}_{\mathrm{SRS}} = \frac{\gamma_{\mathrm{SRS}}}{(1-\delta) f(\xi)}$.

4.5 Asymptotic Analysis of i.i.d. Sum

We now provide a theoretical comparison of the EC estimators from Sections 4.2–4.4, showing that MSIS (Section 4.4.1) can outperform the other methods. Our study considers a loss Y as the sum of m i.i.d. random variables (i.e., a random walk, often a building block in more complex models) in an asymptotic regime of [41], where $m \to \infty$ with the quantile level p simultaneously approaching 1; see Equation (4.29) below. In addition to its theoretical convenience, the framework also has practical relevance: bank portfolios are commonly exposed to thousands of obligors (i.e., large m), and as noted in Section 1.3, [24, p. 46], e.g., has used $p = 0.999$ and $p = 0.9998$ in its EC computations. Although the analysis in this section is for an i.i.d. sum model, the dependent sum in the more complicated PCRM (Section 4.6.2) can be reduced to an independent (but not necessarily identically distributed) sum through appropriate conditioning arguments, as in [40] and [15]. Thus, the asymptotics for the i.i.d. sum provide insights about how exponential twisting may behave for factor models with dependence.

Assumption 1. *The loss in Equation* (4.1) *is* $Y = c(\mathbf{X}) = \sum_{j=1}^m X_j$ *for* $d = m$*, where* $\mathbf{X} = (X_1, X_2, \ldots, X_m) \sim G$ *has i.i.d. components. Each* X_j *has marginal CDF* G_0 *that does not depend on m, mean* $\mu_0 \equiv E_0[X_j] = \int x\, dG_0(x)$*, and variance* $\sigma_0^2 \equiv Var_0[X_j] > 0$*,*

where E_0 and Var_0 are the expectation and variance operators, respectively, when $X_j \sim G_0$. The moment generating function (MGF) $M_0(\theta) = E_0[e^{\theta X_j}] = \int e^{\theta x} dG_0(x)$ of $X_j \sim G_0$ has domain $\Delta = \{\theta \in \mathbb{R} : M_0(\theta) < \infty\}$ with interior Δ° containing 0. SRS samples i.i.d. copies of $c(\mathbf{X})$ with $\mathbf{X} \sim G$, and the computation (CPU) time to generate $c(\mathbf{X})$ with $\mathbf{X} \sim G$ is a random variable (a constant being a special case) with expectation $m\tau_{SRS}$ for some constant $\tau_{SRS} \in (0, \infty)$. The quantile level p satisfies

$$p \equiv p_m = 1 - e^{-\beta m}, \text{ for some constant } \beta > 0, \tag{4.29}$$

and the derivative f of the CDF F of Y exists at $\xi = F^{-1}(p)$, with $f(\xi) > 0$.

The domain Δ of M_0 always contains 0, but its interior Δ° may not, as for heavy-tailed distributions, e.g., lognormal or Pareto [9, Section VI.3]. Thus, the condition that $0 \in \Delta^\circ$ in Assumption 1 restricts us to light-tailed summands [9, Section VI.2], e.g., normal or gamma. In this case [17, p. 278], all moments of $X_j \sim G_0$ are finite, and $M_0(\theta)$ has derivatives of all orders for $\theta \in \Delta^\circ$; let $M_0'(\theta) = \frac{\mathrm{d}}{\mathrm{d}\theta} M_0(\theta)$ and $M_0''(\theta) = \frac{\mathrm{d}^2}{\mathrm{d}\theta^2} M_0(\theta)$. Define $Q_0(\theta) = \ln M_0(\theta)$ as the *cumulant generating function* (CGF) of $X_j \sim G_0$, with $Q_0'(\theta) = \frac{\mathrm{d}}{\mathrm{d}\theta} Q_0(\theta)$ and $Q_0''(\theta) \frac{\mathrm{d}^2}{\mathrm{d}\theta^2} Q_0(\theta)$.

We sometimes (but not always) emphasize the dimension m of $\mathbf{X}$ by writing $\eta = \eta_m$ for the EC, $F = F_m$ as the CDF of Y, $\xi = \xi_m = F_m^{-1}(p)$ as the p-quantile of Y, and $\mu = \mu_m = E_G[Y] = E_G[c(\mathbf{X})]$. Some of our asymptotic analysis will account for the computational effort to construct an EC estimator, in which case our study will incorporate the expected time $m\tau_{\mathrm{SRS}}$ to generate $c(\mathbf{X})$ with $\mathbf{X} \sim G$ from Assumption 1.

4.5.1 Importance Sampling via Exponential Twisting

A common IS approach applies *exponential twisting*, also called *exponential tilting* or an *exponential change of measure* [9, Section VI.2]. We next describe methods IS(θ), ISDM(θ), MSIS(θ), and DE(θ), which utilize twisting with parameter θ in various ways.

Assumption 2. *Under IS(θ), random vector* $\mathbf{X}$ *has joint CDF* $\widetilde{G}_\theta$ *such that the components* $X_1,\ldots,X_m$ *of* $\mathbf{X}$ *are i.i.d., where the marginal CDF of each* X_j *is the exponential twist* $\widetilde{G}_{0,\theta}$ *of* G_0*, given by*

$$d\widetilde{G}_{0,\theta}(x) = \frac{e^{\theta x}\, dG_0(x)}{M_0(\theta)} = e^{\theta x - Q_0(\theta)}\, dG_0(x), \quad x \in \mathbb{R}, \quad \theta \in \Delta^\circ; \tag{4.30}$$

for LR $L_\theta(\mathbf{x}) \equiv dG(\mathbf{x})/d\widetilde{G}_\theta(\mathbf{x})$ *in* (4.11)*, the expected time to generate* $(c(\mathbf{X}), L_\theta(\mathbf{X}))$*,* $\mathbf{X} \sim \widetilde{G}_\theta$*, is* $m\tau_{IS(\theta)}$ *for a constant* $\tau_{IS(\theta)} \in (0,\infty)$*. ISDM*$(\theta) \equiv$ *ISDM*(θ,δ) *is IS with* $\mathbf{X} \sim \widetilde{G}_{ISDM(\theta)} \equiv \delta\widetilde{G}_\theta + (1-\delta)G$ *and fixed* $\delta \in (0,1)$*; for LR* $L_{ISDM(\theta)}(\mathbf{x}) \equiv dG(\mathbf{x})/d\widetilde{G}_{ISDM(\theta)}(\mathbf{x})$ *in Equation* (4.24)*, the expected time to generate* $(c(\mathbf{X}), L_{ISDM(\theta)}(\mathbf{X}))$*,* $\mathbf{X} \sim \widetilde{G}_{ISDM(\theta)}$*, is* $m\tau_{ISDM(\theta)}$ *for a constant* $\tau_{ISDM(\theta)} \in (0,\infty)$*. MSIS*$(\theta) \equiv$ *MSIS*(θ,δ) *and DE*$(\theta) \equiv$ *DE*$(\theta,\delta,\upsilon_1,\upsilon_2)$ *use IS*(θ) *as their IS, with fixed* $\delta \in (0,1)$*, and fixed* $\upsilon_1,\upsilon_2 \in (0,1)$*. There exists* $\theta = \theta_\star \in \Delta^\circ$ *with* $\theta_\star > 0$ *such that*

$$-\theta_\star Q_0'(\theta_\star) + Q_0(\theta_\star) = -\beta, \quad \text{for } \beta > 0 \text{ in Equation (4.29).} \tag{4.31}$$

Reference [41] employs IS$(\theta_\star)$ to estimate the p-quantile ξ in our asymptotic regime with $m \to \infty$ and p as in (4.29), showing that if there is a $\theta_\star > 0$ solving (4.31), it is unique. For each $\beta > 0$, such a $\theta_\star \equiv \theta_\star(\beta) > 0$ exists when, e.g., G_0 is normal or gamma, but there are cases of G_0 and $\beta > 0$ when (4.31) has no root. For example, if G_0 is the "perverted exponential" [29, p. 74], which has density $g_0(x) = c_0 x^{-3} e^{-x} I(x \geq 1)$ and $\Delta = \{\theta \leq 1\}$, where $c_0 \equiv 1/\int_1^\infty x^{-3}e^{-x}\,\mathrm{d}x \doteq 9.116$, then it can be shown that there exists $\theta_\star > 0$ with $\theta_\star \in \Delta^\circ$ solving (4.31) if and only if $0 < \beta < \ln(c_0/2) - 2 \doteq 0.483$.

4.5.2 Relative Error and Work-Normalized Relative Error

We will compare our EC estimators in terms of their relative errors (e.g., [72]). We explain this idea in a general context of a Monte Carlo method $\mathfrak{M}$ (e.g., SRS, IS(θ), MSIS(θ), ISDM(θ), or DE(θ)) for an estimand $\varphi \equiv \varphi_m$ (e.g., η, ξ, or μ) of a sequence of stochastic models indexed by a parameter m (e.g., dimension of $\mathbf{X}$). Let $\widehat{\varphi}_{\mathfrak{M},n} \equiv \widehat{\varphi}_{\mathfrak{M},n,m}$ be the $\mathfrak{M}$

estimator of φ based on a total sample size n. For each fixed m, assume the estimator obeys a CLT $\sqrt{n}[\widehat{\varphi}_{\mathfrak{M},n} - \varphi] \Rightarrow N(0, \omega^2_{\mathfrak{M}})$ as $n \to \infty$, where $\omega^2_{\mathfrak{M}} \equiv \omega^2_{\mathfrak{M},m} < \infty$ is the asymptotic variance. When $\varphi \neq 0$, we define the *relative error* (RE) of the $\mathfrak{M}$ estimator of φ as

$$\mathrm{RE}_{\mathfrak{M},m}[\varphi] = \frac{\omega_{\mathfrak{M}}}{|\varphi|} \equiv \frac{\omega_{\mathfrak{M},m}}{|\varphi_m|}, \tag{4.32}$$

which we will study as $m \to \infty$. (Our definition of RE ignores that $\widehat{\varphi}_{\mathfrak{M},n}$ may be biased, as is often the case when $\varphi = \xi$ or $\varphi = \eta$. But when applying SRS with fixed dimension m, the simplification is reasonable because as $n \to \infty$, the SRS quantile estimator's mean-squared error is determined primarily by its asymptotic variance, with negligible contribution from the bias [10, Theorem 2].)

To motivate the study of RE, consider a 95% confidence interval $(\widehat{\varphi}_{\mathfrak{M},n} \pm 1.96\omega_{\mathfrak{M}}/\sqrt{n})$ for φ based on the CLT for $\widehat{\varphi}_{\mathfrak{M},n}$. (In practice, $\omega_{\mathfrak{M}}$ is typically unknown and is replaced by a consistent estimator.) Suppose that we want to determine a sample size n so that the CI is roughly $(\widehat{\varphi}_{\mathfrak{M},n} \pm \varepsilon|\widehat{\varphi}_{\mathfrak{M},n}|)$ for a specified desired relative precision $\varepsilon > 0$; e.g., if $\varepsilon = 0.1$, then the desired CI has 10% relative half-width. Thus, we seek n so that $1.96\omega_{\mathfrak{M}}/\sqrt{n} \approx \varepsilon|\varphi|$, or equivalently, $n \approx (1.96\,\mathrm{RE}_{\mathfrak{M},m}[\varphi]/\varepsilon)^2$. If $\mathrm{RE}_{\mathfrak{M},m}[\varphi]$ is bounded (resp., grows to ∞) as $m \to \infty$, then the sample size n needed to achieve a fixed relative precision ε remains bounded (resp., blows up) as m increases. [72] and [9, Chapter VI] review a variety of simulation methods $\mathfrak{M}$ that achieve the desirable property of bounded or even vanishing RE when estimating some parameter φ for various stochastic models and asymptotic regimes.

As m grows, the computation (CPU) time to generate one output for method $\mathfrak{M}$ often increases with m. For example, Assumption 1 specifies $m\tau_{\mathrm{SRS}}$ as the expected CPU time to generate an SRS output $c(\mathbf{X})$ for $\mathbf{X} \in \mathbb{R}^m$ with $\mathbf{X} \sim G$, and Assumption 2 imposes similar structure for $\mathrm{IS}(\theta)$ and $\mathrm{ISDM}(\theta)$. For a method $\mathfrak{M}$ that estimates φ through a *single* i.i.d. sample of size n (as for SRS, $\mathrm{IS}(\theta)$, and $\mathrm{ISDM}(\theta)$), let $m\tau_{\mathfrak{M}}$ be the expected CPU time to generate one output, with $\tau_{\mathfrak{M}} > 0$ a constant. To account for the CPU time for such a

method $\mathfrak{M}$, we define the *work-normalized RE* (WNRE) of the $\mathfrak{M}$ estimator of $\varphi \neq 0$ as

$$\mathrm{WNRE}_{\mathfrak{M},m}[\varphi] = \frac{\sqrt{m\tau_{\mathfrak{M}}}\,\omega_{\mathfrak{M}}}{|\varphi|} \equiv \frac{\sqrt{m\tau_{\mathfrak{M}}}\,\omega_{\mathfrak{M},m}}{|\varphi_m|} = \sqrt{m\tau_{\mathfrak{M}}}\,\mathrm{RE}_{\mathfrak{M},m}[\varphi]; \tag{4.33}$$

see [72]. To motivate the WNRE, suppose that we have a CPU budget $b_0 > 0$. Within our budget b_0, method $\mathfrak{M}$ obtains a sample of size approximately $n_{\mathfrak{M},b_0} \equiv \lfloor b_0/(m\tau_{\mathfrak{M}})\rfloor$. When $n_{\mathfrak{M},b_0} \geq 1$, the resulting $\mathfrak{M}$ estimator of φ based on budget b_0 is then roughly $\widehat{\varphi}_{\mathfrak{M},n_{\mathfrak{M},b_0}}$, whose variance is approximately $\omega^2_{\mathfrak{M}}/n_{\mathfrak{M},b_0} \approx m\tau_{\mathfrak{M}}\omega^2_{\mathfrak{M}}/b_0$, which we can express through the CLT $\sqrt{b_0}[\widehat{\varphi}_{\mathfrak{M},n_{\mathfrak{M},b_0}} - \varphi] \Rightarrow N(0, m\tau_{\mathfrak{M}}\omega^2_{\mathfrak{M}})$ as $b_0 \to \infty$. (We can formalize this argument through a random-time-change CLT, e.g., see [17, p. 369].) Thus, the budget-constrained estimator's standard deviation is roughly scaled by the square root of the expected time to generate one output, which leads to the definition of the WNRE in Equation (4.33).

While Equation (4.33) is appropriate when $\mathfrak{M}$ utilizes only a single i.i.d. sample, MSIS(θ) and DE(θ) instead collect *multiple* samples, and we will define their WNRE by slightly adjusting how these estimators are constructed. Consider estimating $\varphi = \mu$ or $\varphi = \xi$ via DE(θ). (Section 4.5.3 will explain how to handle $\varphi = \eta$ and MSIS(θ); see Equations (4.60) and (4.61).) DE(θ) takes two independent samples: one with IS(θ) and the other with SRS. Rather than dividing the total *sample size* n between IS(θ) and SRS using allocation parameter $0 < \delta < 1$, as in the DE(θ) estimator in Equation (4.25), we instead split the *CPU budget* b_0 when considering WNRE, where δb_0 (resp., $(1-\delta)b_0$) of the budget is for IS(θ) (resp., SRS). Then the IS(θ) (resp., SRS) sample has approximately size $n'_{b_0,1} \equiv \lfloor \delta b_0/(m\tau_{\mathrm{IS}(\theta)})\rfloor$ (resp, $n'_{b_0,2} \equiv \lfloor (1-\delta)b_0/(m\tau_{\mathrm{SRS}})\rfloor$), so the variance of the budget-constrained IS(θ) (resp., SRS) estimator of φ is roughly $\omega^2_{\mathrm{IS}(\theta)}/n'_{b_0,1}$ (resp., $\omega^2_{\mathrm{SRS}}/n'_{b_0,2}$). We form the budget-constrained DE(θ) estimator of φ as a weighted average of the budget-constrained IS(θ) and SRS estimators of ϕ using respective weights υ and $\upsilon' = 1-\upsilon$, where $\upsilon = \upsilon_1$ when $\varphi = \xi$, and $\upsilon = \upsilon_2$ when $\varphi = \mu$, with υ_1 and υ_2 as in Equation (4.25). As DE(θ) applies IS(θ) and SRS independently, the variance of the budget-constrained DE(θ) estimator of φ is roughly $\frac{\upsilon^2\omega^2_{\mathrm{IS}(\theta)}}{n'_{b_0,1}} + \frac{\upsilon'^2\omega^2_{\mathrm{SRS}}}{n'_{b_0,2}} \approx$

$\frac{m}{b_0}\left[\frac{\tau_{\mathrm{IS}(\theta)}\upsilon^2\omega^2_{\mathrm{IS}(\theta)}}{\delta}+\frac{\tau_{\mathrm{SRS}}\upsilon'^2\omega^2_{\mathrm{SRS}}}{1-\delta}\right]$. Thus, we define the WNRE for the DE(θ) estimator of $\varphi\neq 0$ for $\varphi=\xi$ or μ as

$$\mathrm{WNRE}_{\mathrm{DE}(\theta),m}[\varphi]=\frac{1}{|\varphi|}\left[m\left(\frac{\tau_{\mathrm{IS}(\theta)}\upsilon^2\omega^2_{\mathrm{IS}(\theta)}}{\delta}+\frac{\tau_{\mathrm{SRS}}\upsilon'^2\omega^2_{\mathrm{SRS}}}{1-\delta}\right)\right]^{1/2}. \tag{4.34}$$

While we can derive the exact RE and WNRE for estimators of $\varphi=\mu$, our analyses for $\varphi=\xi$ or $\varphi=\eta$ will apply approximations to φ and asymptotic variance $\omega^2_{\mathfrak{M}}$, as will be explained in Sections 4.5.3–4.5.3. Numerical results in Sections 4.6.1 and A will demonstrate that our approximations are quite reasonable.

4.5.3 (Approximate) RE and WNRE for Estimators of μ, ξ, and η

To give limiting results (as $m\to\infty$) about our estimators of μ, ξ, and η in Theorems 6–8 below, we adopt the following asymptotic notation. For functions $r_1(m)$ and $r_2(m)$, we write $r_1(m)=O(r_2(m))$ (resp., $r_1(m)=\Omega(r_2(m))$) as $m\to\infty$ if there are constants $\dot{c},m_0>0$ such that $|r_1(m)|\leq\dot{c}|r_2(m)|$ (resp., $|r_1(m)|\geq\dot{c}|r_2(m)|$) for all $m\geq m_0$, so $\dot{c}|r_2(m)|$ provides an asymptotic *upper* (resp., *lower*) bound for $|r_1(m)|$. Also, $r_1(m)=o(r_2(m))$ means $r_1(m)/r_2(m)\to 0$ as $m\to\infty$.

Estimating μ First consider (4.32)–(4.34) with $\varphi=\mu=E_G[c(\mathbf{X})]=E_{\widetilde{G}_\theta}[c(\mathbf{X})L_\theta(\mathbf{X})]=m\mu_0$. We want to analyze the asymptotic behavior (as $m\to\infty$) of estimators of μ for methods $\mathfrak{M}=$ SRS (also used by MSIS(θ)), IS(θ), ISDM(θ), and DE(θ). The next result, proven in Appendix F, provides expressions and bounds for the estimators' exact variances, the RE in Equation (4.32), and the WNRE in Equations (4.33)–(4.34).

Theorem 6 Under Assumption 1, the following hold for method-$\mathfrak{M}$ estimators of $\mu=E_G[c(\mathbf{X})]$.

(i) The $\mathfrak{M}=$ SRS estimator $\widehat{\mu}_{\mathrm{SRS},n}$ in Equation (4.3) has variance $=\sigma^2_{\mathrm{SRS}}/n$, with

$$\sigma^2_{\mathrm{SRS}}=\mathrm{Var}_G[c(\mathbf{X})]=\sigma_0^2 m, \tag{4.35}$$

so when $\mu_0 \neq 0$,

$$\mathrm{RE}_{\mathrm{SRS},m}[\mu] = \left(\frac{\sigma_0}{|\mu_0|}\right)\frac{1}{\sqrt{m}} \to 0 \text{ as } m \to \infty, \text{ and } \mathrm{WNRE}_{\mathrm{SRS},m}[\mu] = \frac{\sqrt{\tau_{\mathrm{SRS}}}\,\sigma_0}{|\mu_0|} \text{ for all } m. \tag{4.36}$$

All of the remaining parts further impose Assumption 2, so $\theta \in \Delta^\circ$, and also assume $-\theta \in \Delta^\circ$. In particular, the following hold for $\theta = \theta_\star > 0$ solving Equation (4.31) when $\pm\theta_\star \in \Delta^\circ$.

(ii) The $\mathfrak{M} = \mathrm{IS}(\theta)$ estimator $\widehat{\mu}_{\mathrm{IS}(\theta),n}$ in Equation (4.12) has variance $\sigma^2_{\mathrm{IS}(\theta)}/n$, with

$$\sigma^2_{\mathrm{IS}(\theta)} \equiv \mathrm{Var}_{\widetilde{G}_\theta}[c(\mathbf{X})L_\theta(\mathbf{X})] = \begin{cases} M_0(\theta)M_0''(-\theta) - \mu_0^2 & \text{for } m = 1, \\ m[\alpha(\theta)]^m\Big(m[Q_0'(-\theta)]^2 + Q_0''(-\theta)\Big) - (m\mu_0)^2 & \text{for } m \geq 2, \end{cases} \tag{4.37}$$

$$\text{where} \quad \alpha(\theta) \equiv M_0(\theta)M_0(-\theta) \geq 1 \quad \text{and} \quad Q_0''(-\theta) > 0. \tag{4.38}$$

If $\theta \neq 0$, then $\alpha(\theta) > 1$, so $\sigma^2_{\mathrm{IS}(\theta)} = \Omega(m[\alpha(\theta)]^m)$ as $m \to \infty$. If in addition $\mu_0 \neq 0$, then as $m \to \infty$,

$$\begin{aligned} \mathrm{RE}_{\mathrm{IS}(\theta),m}[\mu] &= \Omega([\alpha(\theta)]^{m/2}/\sqrt{m}) \to \infty, \\ \text{and } \mathrm{WNRE}_{\mathrm{IS}(\theta),m}[\mu] &= \Omega([\alpha(\theta)]^{m/2}) \to \infty. \end{aligned} \tag{4.39}$$

(iii) The $\mathfrak{M} = \mathrm{ISDM}(\theta)$ estimator $\widehat{\mu}_{\mathrm{ISDM}(\theta),n}$ in Equation (4.12) has variance $\sigma^2_{\mathrm{ISDM}(\theta)}/n$, with

$$\sigma^2_{\mathrm{ISDM}(\theta)} \equiv \mathrm{Var}_{\widetilde{G}_{\mathrm{ISDM}(\theta)}}[c(\mathbf{X})L_{\mathrm{ISDM}(\theta)}(\mathbf{X})] \leq \frac{\delta\mu_0^2}{1-\delta}m^2 + \frac{\sigma_0^2}{1-\delta}m, \tag{4.40}$$

$$\text{so} \quad \mathrm{RE}_{\mathrm{ISDM}(\theta),m}[\mu] = O(1) \text{ and } \mathrm{WNRE}_{\mathrm{ISDM}(\theta),m}[\mu] = O(\sqrt{m}) \text{ as } m \to \infty. \tag{4.41}$$

(iv) For $\theta \neq 0$, the $\mathfrak{M} = \mathrm{DE}(\theta)$ estimator $\widehat{\mu}_{\mathrm{DE}(\theta),n}$ in Equation (4.25) has variance $\sigma^2_{\mathrm{DE}(\theta)}/n$, with

$$\sigma^2_{\mathrm{DE}(\theta)} \equiv \frac{\upsilon_2^2}{\delta}\sigma^2_{\mathrm{IS}} + \frac{\upsilon_2'^2}{1-\delta}\sigma^2_{\mathrm{SRS}} = \Omega(m[\alpha(\theta)]^m) \quad \text{as } m \to \infty, \tag{4.42}$$

$$\text{so } \mathrm{RE}_{\mathrm{DE}(\theta),m}[\mu] = \Omega([\alpha(\theta)]^{m/2}/\sqrt{m}) \to \infty \text{ and } \mathrm{WNRE}_{\mathrm{DE}(\theta),m}[\mu] = \Omega([\alpha(\theta)]^{m/2}) \to \infty. \tag{4.43}$$

Theorem 6 shows that when we estimate μ via SRS or ISDM(θ), the variance, RE, and WNRE behave polynomially in m as $m \to \infty$, by Equations (4.35), (4.36), (4.40), and (4.41). But IS(θ) with *any* fixed $\theta \neq 0$ results in exponential growth, by Equations (4.37) and (4.39). As seen in Equation (4.42), DE(θ) takes on the asymptotic characteristics of the *worse* of SRS and IS(θ). For some stochastic models of fixed dimension m, [50], who proves Equation (4.40), provides numerical/simulation results showing that using an IS method designed to estimate a tail probability leads to poor mean estimators compared to SRS. Our Theorem 6(ii) provides rigorous supporting theory for the complementary setting of a sum of m i.i.d. random variables as $m \to \infty$.

Estimating ξ Next take φ in Equations (4.32)–(4.34) as the p-quantile $\xi = F^{-1}(p)$. For the methods $\mathfrak{M}$ in Assumptions 1 and 2, the asymptotic variance $\omega^2_{\mathfrak{M}} = \kappa^2_{\mathfrak{M}}$ of the resulting estimator $\widehat{\xi}_{\mathfrak{M},n}$ has the form

$$\kappa^2_{\mathfrak{M}} = \frac{\chi^2_{\mathfrak{M}}}{f^2(\xi)} \tag{4.44}$$

when $f(\xi) > 0$ (Assumption 1). The $\mathfrak{M} = \mathrm{SRS}$ estimator $\widehat{\xi}_{\mathfrak{M},n} = \widehat{\xi}_{\mathrm{SRS},n}$ in Equation (4.4) has $\kappa^2_{\mathfrak{M}} = \tau^2_{\mathrm{MC}}$ with $\chi^2_{\mathfrak{M}} = \chi^2_{\mathrm{SRS}} \equiv \mathrm{Var}_G[I(c(\mathbf{X}) > \xi)] = p(1-p)$, as in Equation (4.10); see [117, Section 2.3.3]. The $\mathfrak{M} = \mathrm{IS}(\theta_\star)$ estimator $\widehat{\xi}_{\mathfrak{M},n} = \widehat{\xi}_{\mathrm{IS}(\theta_\star),n}$ in Equation (4.14) has $\kappa^2_{\mathfrak{M}} = \kappa^2_{\mathrm{IS}(\theta_\star)}$ with $\chi^2_{\mathfrak{M}} = \chi^2_{\mathrm{IS}(\theta_\star)} \equiv \mathrm{Var}_{\widetilde{G}_{\theta_\star}}[L(\mathbf{X})I(c(\mathbf{X}) > \xi)]$, as in Equation (4.19); see [41]. Theorem 7 below defines $\chi^2_{\mathfrak{M}}$ for $\mathfrak{M} = \mathrm{ISDM}(\theta_\star)$ and DE$(\theta_\star)$.

We now want to study the RE and WNRE of the SRS, IS($\theta_\star$) (also used by MSIS($\theta_\star$)), ISDM($\theta_\star$), and DE($\theta_\star$) estimators of $\xi = F^{-1}(p)$ when $F \equiv F_m$ as $m \to \infty$ for quantile level $p \equiv p_m$ as in Equation (4.29). [41] analyzes (only) the numerator $\chi^2_{\mathfrak{M}}$ in Equation (4.44) for $\mathfrak{M} =$ SRS and IS($\theta_\star$) in this asymptotic regime, proving that $\lim_{m\to\infty}(1/m)\ln(\chi^2_{\text{SRS}}) = -\beta$ and $\lim_{m\to\infty}(1/m)\ln(\chi^2_{\text{IS}(\theta_\star)}) \le -2\beta$. This indicates that IS($\theta_\star$) can produce substantial variance reductions.

Further analyzing the quantile estimators' RE and WNRE requires understanding the asymptotic properties of ξ and the denominator $f^2(\xi)$ in Equation (4.44). But the exact values of $\xi \equiv \xi_m$ and $f(\xi) \equiv f_m(\xi_m)$ are usually analytically intractable for large m, so we apply approximations. Specifically, we approximate ξ by

$$\breve{\xi} \equiv \breve{\xi}_m = mQ_0'(\theta_\star), \quad \text{which satisfies} \quad \frac{\breve{\xi}_m - \xi_m}{m} \to 0 \text{ as } m \to \infty; \tag{4.45}$$

see Theorem 2 in [41]. (For a generic quantity φ, we let $\breve{\varphi}$ be a non-simulation approximation.) Also, for studying $f(\xi)$, the exact CDF F of the i.i.d. sum $c(\mathbf{X}) = \sum_{j=1}^m X_j$ is the m-fold convolution of G_0, but such a convolution for F (and f) is often intractable for large m. Instead, we employ a *saddlepoint approximation* [53, Chapter 2] to $f(x)$, $x \in \mathbb{R}$, given by (assuming θ_x below exists)

$$\breve{f}(x) \equiv \breve{f}_m(x) = \frac{1}{\sqrt{2\pi m Q_0''(\theta_x)}} \exp\left[mQ_0(\theta_x) - x\theta_x\right], \text{ for } \theta_x \in \Delta^\circ \text{ satisfying } mQ_0'(\theta_x) = x. \tag{4.46}$$

(Section 4.6.1 and Appendix A will present numerical results for some particular summand CDFs G_0 showing that these approximations and others we will develop in Section 4.5.3 are quite reasonable.)

We now use (4.45) and (4.46) to approximate $\kappa^2_{\mathfrak{M}}$ in (4.44) for $\mathfrak{M} =$ SRS, IS($\theta_\star$) (also used by MSIS($\theta_\star$)), ISDM($\theta_\star$), and DE($\theta_\star$). Replacing $f(\xi)$ by $\breve{f}(\breve{\xi})$, which is

always positive, we will approximate $\kappa^2_{\mathfrak{M}}$ via

$$\breve{\kappa}^2_{\mathfrak{M}} \equiv \frac{\breve{\chi}^2_{\mathfrak{M}}}{\breve{f}^2(\breve{\xi})} \tag{4.47}$$

in Theorem 7 below. For $\breve{\xi} \neq 0$, we approximate $\mathrm{RE}_{\mathfrak{M},m}[\xi] = \frac{\kappa_{\mathfrak{M}}}{|\xi|}$ and $\mathrm{WNRE}_{\mathfrak{M},m}[\xi] = \sqrt{m\tau_{\mathfrak{M}}}\,\mathrm{RE}_{\mathfrak{M},m}$ by

$$\mathrm{R\breve{E}}_{\mathfrak{M},m}[\xi] \equiv \frac{\breve{\kappa}_{\mathfrak{M}}}{|\breve{\xi}|} \quad \text{and} \quad \mathrm{WNR\breve{E}}_{\mathfrak{M},m}[\xi] \equiv \sqrt{m\tau_{\mathfrak{M}}}\,\mathrm{R\breve{E}}_{\mathfrak{M},m}[\xi] \tag{4.48}$$

when $\mathfrak{M} \neq \mathrm{DE}(\theta_\star)$. For $\mathrm{DE}(\theta_\star)$, we define $\mathrm{WNR\breve{E}}_{\mathrm{DE}(\theta_\star),m}[\xi]$ as in (4.34) but with $\breve{\kappa}^2_{\mathfrak{M}'}$ replacing $\omega^2_{\mathfrak{M}'} = \kappa^2_{\mathfrak{M}'}$ for $\mathfrak{M}' = \mathrm{IS}(\theta_\star)$ and SRS. The next result, proven in Appendix G, analyzes our quantile estimators.

Theorem 7 Under Assumptions 1 and 2, the following hold for method-$\mathfrak{M}$ estimators of $\xi = F^{-1}(p)$, where $\beta > 0$ is from Equation (4.29) and $\theta_\star$ from Equation (4.31) satisfies $Q_0''(\theta_\star) > 0$.

(i) For the $\mathfrak{M} = \mathrm{SRS}$ estimator $\widehat{\xi}_{\mathrm{SRS},n}$ in Equation (4.4), the approximation $\breve{\kappa}^2_{\mathrm{SRS}}$ in Equation (4.47) to τ^2_{MC} in Equation (4.44) satisfies

$$\breve{\kappa}^2_{\mathrm{SRS}} \equiv \frac{\chi^2_{\mathrm{SRS}}}{\breve{f}^2(\breve{\xi})} = [2\pi Q_0''(\theta_\star)]m(1-e^{-\beta m})e^{\beta m}, \quad \text{for } \chi^2_{\mathrm{SRS}} = p(1-p) \text{ from (4.10)}, \tag{4.49}$$

so $\breve{\kappa}^2_{\mathrm{SRS}} = \Omega(me^{\beta m})$ as $m \to \infty$. Moreover, if $Q_0'(\theta_\star) \neq 0$ (so $\breve{\xi} \neq 0$), then

$$\mathrm{R\breve{E}}_{\mathrm{SRS},m}[\xi] = \Omega(e^{\beta m/2}/\sqrt{m}) \to \infty \text{ and } \mathrm{WNR\breve{E}}_{\mathrm{SRS},m}[\xi] = \Omega(e^{\beta m/2}) \to \infty, \text{ as } m \to \infty. \tag{4.50}$$

(ii) For the $\mathfrak{M} = \mathrm{IS}(\theta_\star)$ estimator $\widehat{\xi}_{\mathrm{IS}(\theta_\star),n}$ from Equation (4.14), the approximation $\breve{\kappa}^2_{\mathrm{IS}(\theta_\star)}$ in Equation (4.47) to the asymptotic variance $\kappa^2_{\mathrm{IS}(\theta_\star)}$ in Equation (4.44), with

$\chi^2_{\mathrm{IS}(\theta_\star)} = \mathrm{Var}_{\widetilde{G}_{\theta_\star}}[L_{\theta_\star}(\mathbf{X})I(c(\mathbf{X}) > \xi)]$, satisfies

$$\breve{\kappa}^2_{\mathrm{IS}(\theta_\star)} \equiv \frac{\breve{\chi}^2_{\mathrm{IS}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} \leq [2\pi Q_0''(\theta_\star)]m, \quad \text{for } \breve{\chi}^2_{\mathrm{IS}(\theta_\star)} \equiv \mathrm{Var}_{\widetilde{G}_{\theta_\star}}[L_{\theta_\star}(\mathbf{X})I(c(\mathbf{X}) > \breve{\xi})], \tag{4.51}$$

so if $Q_0'(\theta_\star) \neq 0$,

$$\mathrm{R\breve{E}}_{\mathrm{IS}(\theta_\star),m}[\xi] = O(1/\sqrt{m}) \to 0 \text{ and } \mathrm{WNR\breve{E}}_{\mathrm{IS}(\theta_\star),m}[\xi] = O(1) \text{ as } m \to \infty. \tag{4.52}$$

(iii) For the $\mathfrak{M} = \mathrm{ISDM}(\theta_\star)$ estimator $\widehat{\xi}_{\mathrm{ISDM}(\theta_\star),n}$ from Equation (4.14), the approximation $\breve{\kappa}^2_{\mathrm{ISDM}(\theta_\star)}$ in Equation (4.47) to $\kappa^2_{\mathrm{ISDM}(\theta_\star)}$ in Equation (4.44), with $\chi^2_{\mathrm{ISDM}(\theta_\star)} = \mathrm{Var}_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}[L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X})I(c(\mathbf{X}) > \xi)]$, satisfies

$$\breve{\kappa}^2_{\mathrm{ISDM}(\theta_\star)} \equiv \frac{\breve{\chi}^2_{\mathrm{ISDM}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} \leq \frac{2\pi Q_0''(\theta_\star)}{\delta^2}m, \tag{4.53}$$
$$\text{for } \breve{\chi}^2_{\mathrm{ISDM}(\theta_\star)} \equiv \mathrm{Var}_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}[L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X})I(c(\mathbf{X}) > \breve{\xi})],$$

so if $Q_0'(\theta_\star) \neq 0$,

$$\mathrm{R\breve{E}}_{\mathrm{ISDM}(\theta_\star),m}[\xi] = O(1/\sqrt{m}) \to 0 \text{ and } \mathrm{WNR\breve{E}}_{\mathrm{ISDM}(\theta_\star),m}[\xi] = O(1) \text{ as } m \to \infty. \tag{4.54}$$

(iv) For the $\mathfrak{M} = \mathrm{DE}(\theta_\star)$ estimator $\widehat{\xi}_{\mathrm{DE}(\theta_\star),n}$ from Equation (4.25), the approximation $\breve{\kappa}^2_{\mathrm{DE}(\theta_\star)}$ in Equation (4.47) to $\kappa^2_{\mathrm{DE}(\theta_\star)}$ in Equation (4.44), with $\chi^2_{\mathrm{DE}(\theta_\star)} \equiv \frac{\upsilon_1^2}{\delta}\chi^2_{\mathrm{IS}} + \frac{\upsilon_1'^2}{1-\delta}\chi^2_{\mathrm{SRS}}$ as in Equation (4.27), satisfies

$$\breve{\kappa}^2_{\mathrm{DE}(\theta_\star)} \equiv \frac{\breve{\chi}^2_{\mathrm{DE}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} = \Omega(me^{\beta m}) \text{ as } m \to \infty, \quad \text{for } \breve{\chi}^2_{\mathrm{DE}(\theta_\star)} \equiv \frac{\upsilon_1^2}{\delta}\breve{\chi}^2_{\mathrm{IS}} + \frac{\upsilon_1'^2}{1-\delta}\chi^2_{\mathrm{SRS}}. \tag{4.55}$$

Moreover, if $Q_0'(\theta_\star) \neq 0$, then as $m \to \infty$,

$$\mathrm{R\check{E}}_{\mathrm{DE}(\theta_\star),m}[\xi] = \Omega(e^{\beta m/2}/\sqrt{m}) \to \infty, \text{ and } \mathrm{WNR\check{E}}_{\mathrm{DE}(\theta_\star),m}[\xi] = \Omega(e^{\beta m/2}) \to \infty. \tag{4.56}$$

Theorems 6 and 7 show that SRS and IS($\theta_\star$) have opposite effects when estimating μ and ξ. SRS (resp., IS($\theta_\star$)) leads to polynomial (resp., exponential) behavior (in m, as $m \to \infty$) for the (asymptotic) variance, RE, and WNRE when estimating μ (Theorem 6(i)–(ii)), but the estimator of ξ behaves exponentially (resp., polynomially) (Theorem 7(i)–(ii)). For estimating ξ, ISDM($\theta_\star$) inflates the upper bound of the asymptotic variance of IS($\theta_\star$) by a factor of $1/\delta^2$ (compare (4.51) and (4.53)), but as $\delta \in (0,1)$ is fixed, ISDM($\theta_\star$) still has polynomial behavior. By (4.55), DE($\theta_\star$) adopts the limiting characteristics of the *worse* of SRS and IS($\theta_\star$).

Estimating EC We now want to exploit Theorems 6 and 7 to establish results about the approximate RE and WNRE of our EC estimators, which will require defining additional approximations. Using (4.45) and for $\mu_m = m\mu_0$, we approximate the EC $\eta \equiv \eta_m = \xi_m - \mu_m$ by

$$\check{\eta} \equiv \check{\eta}_m \equiv \check{\xi}_m - \mu_m = [Q_0'(\theta_\star) - \mu_0]m, \quad \text{which satisfies} \quad \frac{\check{\eta}_m - \eta_m}{m} \to 0 \text{ as } m \to \infty. \tag{4.57}$$

For method $\mathfrak{M}$ equaling SRS, IS($\theta_\star$), MSIS($\theta_\star$), or ISDM($\theta_\star$), we approximate the asymptotic variance $\zeta_{\mathfrak{M}}^2$ in (4.10), (4.19), and (4.23) of the resulting η estimator by

$$\check{\zeta}_{\mathfrak{M}}^2 = \Lambda_{\mathfrak{M}} \frac{\check{\chi}_{\mathfrak{M}}^2}{\check{f}^2(\check{\xi})} + \Lambda_{\mathfrak{M}}^\dagger \sigma_{\mathfrak{M}}^2 + 2\Lambda_{\mathfrak{M}}^\ddagger \frac{\check{\gamma}_{\mathfrak{M}}}{\check{f}(\check{\xi})} \tag{4.58}$$

with $\check{f}$ from (4.46), and we next define the terms $\check{\chi}_{\mathfrak{M}}^2$, $\sigma_{\mathfrak{M}}^2$, $\check{\gamma}_{\mathfrak{M}}$, $\Lambda_{\mathfrak{M}}$, $\Lambda_{\mathfrak{M}}^\dagger$, and $\Lambda_{\mathfrak{M}}^\ddagger$ in (4.58) for each $\mathfrak{M}$.

- $\mathfrak{M} = \mathrm{SRS}$ has $\check{\chi}_{\mathfrak{M}}^2 = \chi_{\mathrm{SRS}}^2$, $\sigma_{\mathfrak{M}}^2 = \sigma_{\mathrm{SRS}}^2$ from (4.35), $\check{\gamma}_{\mathfrak{M}} = \gamma_{\mathrm{SRS}}$, and $\Lambda_{\mathrm{SRS}} = \Lambda_{\mathrm{SRS}}^\dagger = \Lambda_{\mathrm{SRS}}^\ddagger = 1$, as in (4.10).

- $\mathfrak{M} = \mathrm{IS}(\theta_\star)$ has $\check{\chi}^2_{\mathfrak{M}} = \check{\chi}^2_{\mathrm{IS}(\theta_\star)}$ from (4.51), $\sigma^2_{\mathfrak{M}} = \sigma^2_{\mathrm{IS}(\theta_\star)}$ from (4.37), $\check{\gamma}_{\mathfrak{M}} = \check{\gamma}_{\mathrm{IS}(\theta_\star)} \equiv \mathrm{Cov}_{\widetilde{G}_{\theta_\star}}[I(c(\mathbf{X}) > \check{\xi})L_{\theta_\star}(\mathbf{X}), c(\mathbf{X})L_{\theta_\star}(\mathbf{X})]$ approximating $\gamma_{\mathrm{IS}} = \gamma_{\mathrm{IS}(\theta_\star)}$ in (4.20), and $\Lambda_{\mathrm{IS}(\theta_\star)} = \Lambda^{\dagger}_{\mathrm{IS}(\theta_\star)} = -\Lambda^{\ddagger}_{\mathrm{IS}(\theta_\star)} = 1$, as in (4.19).

- $\mathfrak{M} = \mathrm{MSIS}(\theta_\star)$ has $\check{\chi}^2_{\mathfrak{M}} = \check{\chi}^2_{\mathrm{IS}(\theta_\star)}$ from (4.51), $\sigma^2_{\mathfrak{M}} = \sigma^2_{\mathrm{SRS}}$ from (4.35), $\check{\gamma}_{\mathfrak{M}} = 0$ (ξ and μ are estimated independently), $\Lambda_{\mathrm{MSIS}(\theta_\star)} = 1/\delta$, $\Lambda^{\dagger}_{\mathrm{MSIS}(\theta_\star)} = 1/(1-\delta)$, and $\Lambda^{\ddagger}_{\mathrm{MSIS}(\theta_\star)} = 0$, as in (4.23).

- $\mathfrak{M} = \mathrm{ISDM}(\theta_\star)$ has, as in (4.19), $\check{\chi}^2_{\mathfrak{M}} = \check{\chi}^2_{\mathrm{ISDM}(\theta_\star)}$ from (4.53), $\sigma^2_{\mathfrak{M}} = \sigma^2_{\mathrm{ISDM}(\theta_\star)}$ from (4.40), $\check{\gamma}_{\mathfrak{M}} = \check{\gamma}_{\mathrm{ISDM}(\theta_\star)} \equiv \mathrm{Cov}_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}\left[I(c(\mathbf{X}) > \check{\xi})L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X}), c(\mathbf{X})L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X})\right]$ approximating $\gamma_{\mathrm{ISDM}(\theta_\star)} = \mathrm{Cov}_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}\left[I(c(\mathbf{X}) > \xi)L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X}), c(\mathbf{X})L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X})\right]$ in (4.20), and $\Lambda_{\mathrm{ISDM}(\theta_\star)} = \Lambda^{\dagger}_{\mathrm{ISDM}(\theta_\star)} = -\Lambda^{\ddagger}_{\mathrm{ISDM}(\theta_\star)} = 1$.

For $\mathfrak{M} = \mathrm{DE}(\theta_\star)$, we can also write its asymptotic variance approximation to fit into (4.58), but it is more convenient to treat it differently (to define its WNRĚ below in (4.60)). We approximate $\zeta^2_{\mathrm{DE}(\theta_\star)}$ in (4.27) by

$$\begin{aligned}\check{\zeta}^2_{\mathrm{DE}(\theta_\star)} &= \frac{1}{\delta}\left(\upsilon_1^2\frac{\check{\chi}^2_{\mathrm{IS}(\theta_\star)}}{\check{f}^2(\check{\xi})} + \upsilon_2^2\sigma^2_{\mathrm{IS}(\theta_\star)} - 2\upsilon_1\upsilon_2\frac{\check{\gamma}_{\mathrm{IS}(\theta_\star)}}{\check{f}(\check{\xi})}\right)\\ &+ \frac{1}{1-\delta}\left(\upsilon_1'^2\frac{\chi^2_{\mathrm{SRS}}}{\check{f}^2(\check{\xi})} + \upsilon_2'^2\sigma^2_{\mathrm{SRS}} + 2\upsilon_1'\upsilon_2'\frac{\gamma_{\mathrm{SRS}}}{\check{f}(\check{\xi})}\right)\end{aligned} \tag{4.59}$$

(from rearranging (4.27)), where the approximations in (4.59) are from $\mathfrak{M} = \mathrm{IS}(\theta_\star)$.

For each method $\mathfrak{M}$, we approximate its exact $\mathrm{RE}_{\mathfrak{M},m}[\eta] = \zeta_{\mathfrak{M}}/|\eta|$ by $\mathrm{R\check{E}}_{\mathfrak{M},m}[\eta] = \check{\zeta}_{\mathfrak{M}}/|\check{\eta}|$, if $\eta \neq 0$ and $\check{\eta} \neq 0$. When $\mathfrak{M}$ is SRS, $\mathrm{IS}(\theta_\star)$, or $\mathrm{ISDM}(\theta_\star)$, we approximate the exact $\mathrm{WNRE}_{\mathfrak{M},m}[\eta] = \sqrt{m\tau_{\mathfrak{M}}}\,\mathrm{RE}_{\mathfrak{M},m}[\eta]$ by $\mathrm{WNR\check{E}}_{\mathfrak{M},m}[\eta] = \sqrt{m\tau_{\mathfrak{M}}}\,\mathrm{R\check{E}}_{\mathfrak{M},m}[\eta]$, as in (4.33), which is for methods that use only a single i.i.d. sample.

As $\mathrm{MSIS}(\theta_\star)$ and $\mathrm{DE}(\theta_\star)$ utilize more than a single i.i.d. sample, we slightly adjust the definitions of $\mathrm{WNRE}_{\mathfrak{M},m}$ and $\mathrm{WNR\check{E}}_{\mathfrak{M},m}$ for their η estimators. By similar reasoning used to define $\mathrm{WNRE}_{\mathrm{DE}(\theta),m}[\varphi]$ for $\varphi = \xi$ and μ in (4.34), we approximate the exact

$\mathrm{WNRE}_{\mathrm{DE}(\theta_\star),m}[\eta]$ by $\mathrm{WNR\breve{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$, given by

$$\frac{\left[\frac{m\tau_{\mathrm{IS}(\theta_\star)}}{\delta}\left[\upsilon_1^2\frac{\breve{\chi}^2_{\mathrm{IS}(\theta_\star)}}{\breve{f}^2(\breve{\xi})}+\upsilon_2^2\sigma^2_{\mathrm{IS}(\theta_\star)}-2\upsilon_1\upsilon_2\frac{\breve{\gamma}_{\mathrm{IS}(\theta_\star)}}{\breve{f}(\breve{\xi})}\right]+\frac{m\tau_{\mathrm{SRS}}}{1-\delta}\left[\upsilon_1'^2\frac{\chi^2_{\mathrm{SRS}}}{\breve{f}^2(\breve{\xi})}+\upsilon_2'^2\sigma^2_{\mathrm{SRS}}+2\upsilon_1'\upsilon_2'\frac{\gamma_{\mathrm{SRS}}}{\breve{f}(\breve{\xi})}\right]\right]^{1/2}}{|\breve{\eta}|} \tag{4.60}$$

Similarly, for $\mathrm{MSIS}(\theta_\star)$, we define

$$\mathrm{WNR\breve{E}}_{\mathrm{MSIS}(\theta_\star),m}[\eta]=\frac{1}{|\breve{\eta}|}\left[m\left(\frac{\tau_{\mathrm{IS}(\theta_\star)}\breve{\chi}^2_{\mathrm{IS}(\theta_\star)}}{\delta\breve{f}^2(\breve{\xi})}+\frac{\tau_{\mathrm{SRS}}\sigma^2_{\mathrm{SRS}}}{1-\delta}\right)\right]^{1/2} \tag{4.61}$$

to approximate the exact $\mathrm{WNRE}_{\mathrm{MSIS}(\theta_\star),m}[\eta]$ defined analogously with approximations $(\breve{\eta},\breve{\chi}^2_{\mathrm{IS}(\theta_\star)},\breve{f}^2(\breve{\xi}))$ replaced by their exact values $(\eta,\chi^2_{\mathrm{IS}(\theta_\star)},f^2(\xi))$.

Recall that $(\upsilon_1^*,\upsilon_2^*)$ in (4.28) minimizes ζ^2_{DE} in (4.27) for fixed $\delta\in(0,1)$. Next we define an approximation of $(\upsilon_1^*,\upsilon_2^*)$ for our i.i.d. sum model. Since the value of η does not depend on υ_1 and υ_2, minimizing $\breve{\zeta}^2_{\mathrm{DE},\delta,\upsilon_1,\upsilon_2}$or $\mathrm{R\breve{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$ results in the same approximate optimal value $(\breve{\upsilon}^*_{1,m},\breve{\upsilon}^*_{2,m})=(\frac{\breve{a}_{1,m}}{\breve{a}_{0,m}},\frac{\breve{a}_{2,m}}{\breve{a}_{0,m}})$. Here, $(\breve{a}_{0,m},\breve{a}_{1,m},\breve{a}_{2,m})$ is an approximation of (a_0,a_1,a_2) in (4.28) obtained by replacing $\mathrm{V}^{(\xi)}_{\mathrm{IS}}$ with $\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\equiv\frac{\breve{\kappa}^2_{\mathrm{IS}(\theta_\star)}}{\delta}$ from (4.51), $\mathrm{V}^{(\xi)}_{\mathrm{SRS}}$ with $\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\equiv\frac{\breve{\kappa}^2_{\mathrm{SRS}}}{1-\delta}$ from (4.49), $\mathrm{V}^{(\mu)}_{\mathrm{IS}}$ with $\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\equiv\frac{\sigma^2_{\mathrm{IS}(\theta)}}{\delta}$ from (4.37), $\mathrm{V}^{(\mu)}_{\mathrm{SRS}}$ with $\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m}\equiv\frac{\sigma^2_{\mathrm{SRS}}}{1-\delta}$ from (4.35), C_{IS} with $\breve{\mathrm{C}}_{\mathrm{IS},m}\equiv\frac{\breve{\gamma}_{\mathrm{IS}(\theta_\star)}}{\delta\breve{f}(\breve{\xi})}$ and $\mathrm{C}_{\mathrm{SRS}}$ with $\breve{\mathrm{C}}_{\mathrm{SRS},m}\equiv\frac{\chi^2_{\mathrm{SRS}}}{(1-\delta)\breve{f}^2(\breve{\xi})}$ from (4.59). We also define the approximate optimal value of (υ_1,υ_2) when minimizing $\mathrm{WNR\breve{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$ in the end of Appendix H of the appendices.

Theorem 8 Suppose that Assumptions 1 and 2 hold, so $\theta_\star\in\Delta^\circ$ with $\theta_\star>0$, and further assume $-\theta_\star\in\Delta^\circ$. Then the method-$\mathfrak{M}$ estimators of the EC $\eta=\xi-\mu$ satisfy the following as $m\to\infty$, with $Q_0''(\theta_\star)>0$, $Q_0''(-\theta_\star)>0$, $\beta>0$ from (4.29), and $\alpha(\theta_\star)>1$ from (4.38).

(i) The $\mathfrak{M} = \text{SRS}$ estimator $\widehat{\eta}_{\text{SRS},n}$ in (4.5) has

$$\breve{\zeta}^2_{\text{SRS}} = [2\pi Q_0''(\theta_\star)]m(1-e^{-\beta m})e^{\beta m} + o(m(1-e^{-\beta m})e^{\beta m}) = \Omega(me^{\beta m}), \quad (4.62)$$

$$\text{R}\breve{\text{E}}_{\text{SRS},m}[\eta] = \Omega(e^{\beta m/2}/\sqrt{m}) \to \infty, \text{ and } \text{WNR}\breve{\text{E}}_{\text{SRS},m}[\eta] = \Omega(e^{\beta m/2}) \to \infty. \quad (4.63)$$

(ii) The $\mathfrak{M} = \text{IS}(\theta_\star)$ estimator $\widehat{\eta}_{\text{IS}(\theta_\star),n}$ in (4.17) has

$$\breve{\zeta}^2_{\text{IS}(\theta_\star)} \geq Q_0''(-\theta_\star)m[\alpha(\theta_\star)]^m + o(m[\alpha(\theta_\star)]^m) = \Omega(m[\alpha(\theta_\star)]^m), \quad (4.64)$$

$$\text{R}\breve{\text{E}}_{\text{IS}(\theta_\star),m}[\eta] = \Omega([\alpha(\theta_\star)]^{m/2}/\sqrt{m}) \to \infty, \text{ and } \text{WNR}\breve{\text{E}}_{\text{IS}(\theta_\star),m}[\eta] = \Omega([\alpha(\theta_\star)]^{m/2}) \to \infty. \quad (4.65)$$

(iii) The $\mathfrak{M} = \text{MSIS}(\theta_\star)$ estimator $\widehat{\eta}_{\text{MSIS}(\theta_\star),n}$ in (4.21) has

$$\breve{\zeta}^2_{\text{MSIS}(\theta_\star)} \leq \left(\frac{2\pi Q_0''(\theta_\star)}{\delta} + \frac{\sigma_0^2}{1-\delta}\right)m = O(m), \quad (4.66)$$

$$\text{R}\breve{\text{E}}_{\text{MSIS}(\theta_\star),m}[\eta] = O(1/\sqrt{m}) \to 0, \text{ and } \text{WNR}\breve{\text{E}}_{\text{MSIS}(\theta_\star),m}[\eta] = O(1). \quad (4.67)$$

(iv) The $\mathfrak{M} = \text{ISDM}(\theta_\star)$ estimator $\widehat{\eta}_{\text{ISDM}(\theta_\star),n}$ in (4.17) has

$$\breve{\zeta}^2_{\text{ISDM}(\theta_\star)} \leq \left[\frac{\delta\mu_0^2}{1-\delta}\right]m^2 + \left[\frac{8\pi Q_0''(\theta_\star)}{(1-\delta)}\left(\frac{\mu_0^2}{\delta} + \frac{\sigma_0^2}{m\delta^2}\right)\right]^{1/2} m^{3/2} + \left[\frac{2\pi Q_0''(\theta_\star)}{\delta^2} + \frac{\sigma_0^2}{1-\delta}\right]m$$
$$= O(m^2), \quad (4.68)$$

$$\text{R}\breve{\text{E}}_{\text{ISDM}(\theta_\star),m}[\eta] = O(1), \text{ and } \text{WNR}\breve{\text{E}}_{\text{ISDM}(\theta_\star),m}[\eta] = O(\sqrt{m}). \quad (4.69)$$

(v) The $\mathfrak{M} = \mathrm{DE}(\theta_\star)$ estimator $\widehat{\eta}_{\mathrm{DE}(\theta_\star),n}$ in (4.25) has, for $s_0 \equiv s_0(\theta_\star, \beta) = \max[\alpha(\theta_\star), e^\beta] > 1$,

$$\begin{aligned}\check{\zeta}^2_{\mathrm{DE}(\theta_\star)} &\geq \left[\frac{\upsilon_2^2}{\delta} Q_0''(-\theta_\star)\right] m[\alpha(\theta_\star)]^m + o(m[\alpha(\theta_\star)]^m) + \left[\frac{2\upsilon_1'^2}{1-\delta}\pi Q_0''(\theta_\star)\right] m e^{\beta m} + o(m e^{\beta m}) \\ &= \Omega(m s_0^m), \end{aligned} \tag{4.70}$$

$$\mathrm{R\check{E}}_{\mathrm{DE}(\theta_\star),m}[\eta] = \Omega(s_0^{m/2}/\sqrt{m}) \to \infty, \text{ and } \mathrm{WNR\check{E}}_{\mathrm{DE}(\theta_\star),m}[\eta] = \Omega(s_0^{m/2}) \to \infty, \tag{4.71}$$

for any fixed $\delta, \upsilon_1, \upsilon_2 \in (0,1)$, as in Assumption 2. If instead (υ_1, υ_2) is allowed to depend on m but with $\delta \in (0,1)$ still fixed, the approximate value of $(\upsilon_1^*, \upsilon_2^*)$ in (4.28) satisfies

$$\lim_{m\to\infty} (\check{\upsilon}^*_{1,m}, \check{\upsilon}^*_{2,m}) = (1,0). \tag{4.72}$$

We now sketch Theorem 8's proof, which is in Appendix H. For a method-$\mathfrak{M}$ estimator of $\eta = \xi - \mu$, the growth rate (as $m \to \infty$) of its (approximate) asymptotic variance $\check{\zeta}^2_{\mathfrak{M}}$ in (4.58)–(4.59) is governed by the largest growth rate of the (approximate) variances of its constituent estimators of ξ and μ. (Covariance terms in $\check{\zeta}^2_{\mathfrak{M}}$ are nondominant, by the Cauchy-Schwarz inequality.) Also, $\check{\eta}_m$ grows linearly in m by (4.57).

Applying these insights to SRS, we see that the exponential growth in (4.62) and (4.63) is due to the quantile estimator's exponential increase (Theorem 7(i)). For $\mathrm{IS}(\theta_\star)$, the exponential behavior in (4.64) and (4.65) arises from the mean estimator's exponential rate (Theorem 6(ii)). As $\mathrm{DE}(\theta_\star)$ uses both SRS and $\mathrm{IS}(\theta_\star)$ to estimate both ξ and μ, its η estimator's behavior is determined by the *worst* of those estimators; the base s_0 of the exponential term s_0^m in (4.70)–(4.71) is the *larger* of the SRS base e^β of $(e^\beta)^m$ in (4.62) from Theorem 7(i) and the $\mathrm{IS}(\theta_\star)$ base $\alpha(\theta_\star)$ in (4.64) from Theorem 6(ii).

In contrast, the $\mathrm{MSIS}(\theta_\star)$ (resp., $\mathrm{ISDM}(\theta_\star)$) estimator of $\eta = \xi - \mu$ behaves polynomially in m because the same holds for its constituent estimators of ξ (with $\mathrm{IS}(\theta_\star)$, by Theorem 7(ii)) (resp., Theorem 7(iii)) and μ (with SRS, by Theorem 6(i)) (resp.,

Theorem 6(iii)). Specifically, Theorem 8(iii)–(iv) establish that as $m \to \infty$, the RĚ vanishes (resp., remains bounded) for MSIS$(\theta_\star)$ (resp., ISDM$(\theta_\star)$) by (4.67) (resp., (4.69)), and the WNRĚ for MSIS$(\theta_\star)$ remains bounded by (4.67). While (4.69) shows that an upper bound for the WNRĚ of ISDM$(\theta_\star)$ grows at most at rate $\sqrt{m}$, better than the exponential rates for SRS, IS$(\theta_\star)$ and DE$(\theta_\star)$, this does not necessarily mean that the WNRĚ for ISDM$(\theta_\star)$ blows up, only that its upper bound does. But our numerical results in Sections 4.6.1 and A indicate that the RE and RĚ for ISDM$(\theta_\star)$ flatten out as m increases, so further scaling by $\sqrt{m\tau_{\mathrm{ISDM}(\theta_\star)}}$ would result in WNRE and WNRĚ growing polynomially in m. Finally, as $m \to \infty$, (4.72) shows that when using the optimal approximate DE$(\theta_\star)$ weights to minimize $\check{\zeta}^2_{\mathrm{DE},\delta,\upsilon_1,\upsilon_2}$ and $\mathrm{R\check{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$, the DE$(\theta_\star)$ estimator asymptotically reduces to MSIS$(\theta_\star)$. The end of Appendix H also analyzes this in terms of minimizing $\mathrm{WNR\check{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$ from (4.60), and we get the same result.

4.6 Numerical and Simulation Results

We next present results for two models of a loss $Y = c(\mathbf{X})$ as in (4.1), but with different definitions for c and $\mathbf{X} = (X_1, \ldots, X_d)$. Section 4.6.1 studies the random-walk model in Assumption 1 of Section 4.5, so $c(\mathbf{X}) = \sum_{j=1}^{d} X_j$ has i.i.d. summands with $d = m$; we take the summand CDF G_0 as exponential, whose analytical tractability permits numerical computation. Section 4.6.2 examines a more complicated portfolio credit risk model from (4.2), which we simulate. We compare EC estimators for SRS (Section 4.2), IS (Section 4.3), MSIS (Section 4.4.1), ISDM (Section 4.4.2), and DE (Section 4.4.3). For MSIS, ISDM, and DE, we let $\delta = \upsilon_1 = \upsilon_2 = 1/2$ for all results. For each model, we specify below the joint CDF $\widetilde{G}$ of $\mathbf{X}$ for IS, MSIS, and DE, and we use this same CDF as G^* in ISDM; see (4.24). Although the PCRM is more complex, the results will show that the methods behave similarly on the two models, with MSIS outperforming the other approaches. Thus, our theoretical results (Section 4.5.3) for the i.i.d. sum provide considerable insight into the methods.

4.6.1 Exact and Approximate RE for i.i.d. Sum

As in Assumption 1 of Section 4.5, we define here the loss as $Y = \sum_{j=1}^{m} X_j$, with the X_j as i.i.d. with marginal CDF G_0, and the quantile level $p \equiv p_m$ satisfies (4.29). Theorem 8 establishes that as $m \to \infty$, the approximate relative errors of the SRS, IS($\theta_\star$), and DE($\theta_\star$) estimators of the EC $\eta \equiv \eta_m$ grow exponentially, but MSIS($\theta_\star$) (resp., ISDM($\theta_\star$)) has RĚ that shrinks to 0 (resp., remains bounded). For each EC estimator, we want to investigate numerically the behavior (as m increases) of the *exact* RE (based on (4.10), (4.19), (4.23), or (4.27)) and also compare it to its approximation RĚ. We present here results for G_0 as exponential with mean $\mu_0 = 1$. (Appendix A gives other results when G_0 is $N(1,1)$ and Erlang ($s = 8$ stages).)

The top two rows of Figure 4.1 plot the exact RE of our estimators of η, ξ, and μ as functions of the dimension m. By (4.45) and (4.57), η_m, $\breve{\eta}_m$ ξ_m, $\breve{\xi}_m$, and μ_m share the same growth rate: linear in m. Thus, the RĚ and WNRĚ of η, ξ, and μ are directly comparable, suggesting the same for RE and WNRE. As m grows, the top left panel shows that the SRS, IS($\theta_\star$), and DE($\theta_\star$) estimators of η have exponentially increasing RE, which agree with (4.63), (4.65), and (4.71) for the approximation RĚ. For SRS (resp., IS($\theta_\star$)) the RE of the estimator of η rises exponentially because the same holds for ξ (resp., μ) by (4.50) (resp., (4.39)); see middle panels. Also, as explained after Theorem 8, the RE of the DE($\theta_\star$) estimator of η is governed by the worst of the SRS and IS($\theta_\star$) estimators of ξ and μ, which in this case is the SRS estimator of ξ, as seen in the middle-row panels of Figure 4.1. (For other G_0 in Appendix A, the IS($\theta_\star$) estimator of μ is worst.)

In contrast, the top two rows of Figure 4.1 also show that the MSIS($\theta_\star$) and ISDM($\theta_\star$) estimators of η have decreasing RE as m grows; see (4.67) and (4.69) for RĚ. As m gets large, MSIS($\theta_\star$) appears to be somewhat better than ISDM($\theta_\star$) when estimating η, with MSIS($\theta_\star$) continually decreasing, but ISDM($\theta_\star$) flattening out. The reason becomes apparent from the middle right panel: the estimator of μ using SRS

(which is how MSIS($\theta_\star$) estimates μ) has shrinking RE as m grows (see (4.36)), while the ISDM($\theta_\star$) estimator of μ does not (see (4.41) and the last paragraph of Section 4.5.3).

The bottom-left panel of Figure 4.1 displays the exact RE and its approximation RĚ for the SRS, IS($\theta_\star$), MSIS($\theta_\star$), and ISDM($\theta_\star$) estimators of η, as functions of m. For the same method, the RE and RĚ plots often overlap substantially, especially as m increases, so our approximations can be very accurate. But even when they exhibit some difference for a particular method (as for SRS), the RE and RĚ plots follow the same trends. (We simplified the figure by omitting DE($\theta_\star$); its RE and RĚ for η also behave similarly.) Thus, we conclude that our approximations in Sections 4.5.3 and 4.5.3 are quite reasonable.

4.6.2 Portfolio Credit Risk Model

We next present simulation results from estimating EC on the model in (4.2) of the loss of a credit portfolio with $m = 1000$ obligors and $r = 10$ factors. The model extends one in [40], who assume a constant loss given default for each obligor $k = 1,\dots,m$. We instead let the LGD be a continuous random variable so that F is differentiable at ξ and $f(\xi) > 0$, as required by our theorems.

Recall $\mathbf{X} = (\mathbf{Z}, S, \varepsilon_1, \dots, \varepsilon_m, J_1, \dots, J_m)$ in (4.2) is an $\mathbb{R}^{r+2m+1}$-valued random vector with independent components. As in [40], we take the common shock to be $S \equiv 1$. Let $D_k = I(\mathbf{a}_k\mathbf{Z} + b_k\varepsilon_k > w_k)$ be the indicator function in (4.2) that obligor k defaults. Because $\mathbf{a}_k\mathbf{Z} + b_k\varepsilon_k \sim N(0,1)$, if we set $w_k = \Phi^{-1}(1-p_k)$ for some constant $0 < p_k < 1$, where $\Phi(\cdot)$ is the $N(0,1)$ CDF, then obligor k has marginal default probability $P(D_k = 1) = p_k$. Our experiments used $p_k = 0.01 \cdot (1 + \sin(16\pi k/m))$, $k = 1,\dots,m$, as in [40]. For each obligor $k = 1,2,\dots,m$, the constant LGD in [40] is modified to $C_k = v_k(\mathbf{Z}, S, \varepsilon_1, \dots, \varepsilon_m, J_k) = J_k \sim \text{Unif}(0, \beta_k)$, where $\beta_k = 2 \cdot (\lceil 5k/m \rceil)^2$ and Unif(c_0, c_1) denotes a continuous uniform distribution on (c_0, c_1). As in [40], we randomly generated the loading factors $a_{k,j}$ in (4.2) once as independent Unif$(0, 1/\sqrt{r})$, and used these values in all experiments.

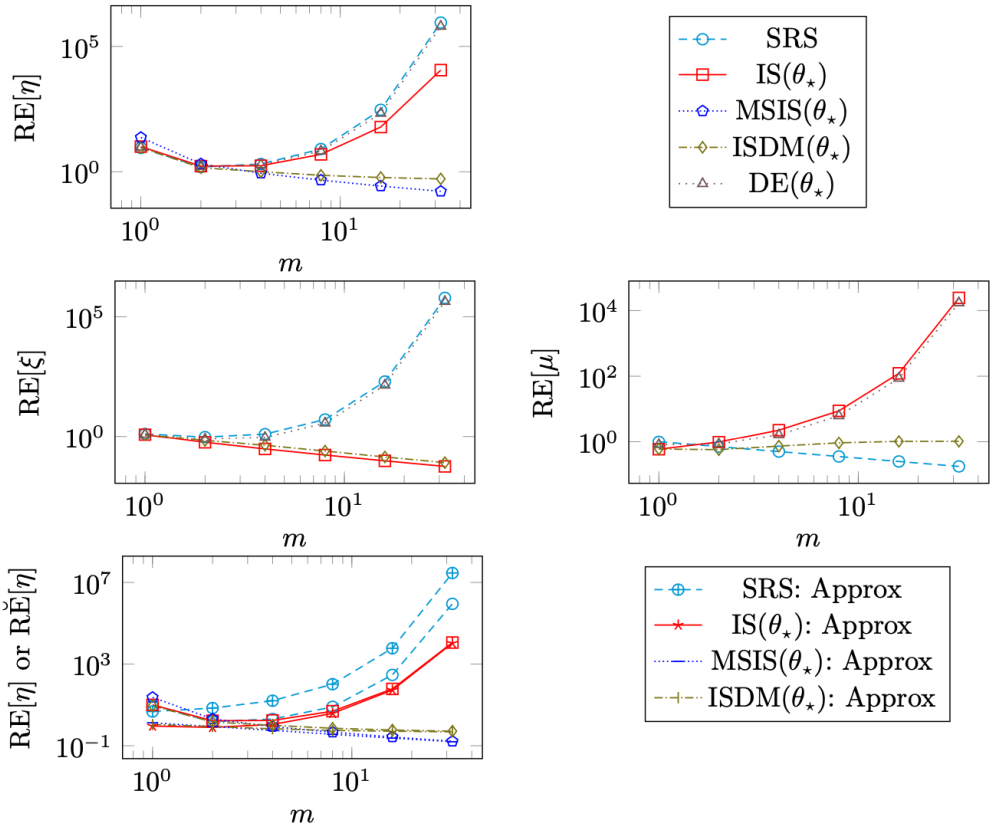


Figure 4.1 For G_0 as exponential (mean 1) and $\beta = 1.1$ in (4.31), the log-log plots show the RE and its approximation RĚ, computed numerically (i.e., not estimated via simulation), as functions of the dimension m. In the top two rows, the plots display the exact RE of estimators of the EC η (top left panel), the p-quantile ξ (middle row, left panel), and the mean μ (middle row, right panel). The bottom left panel shows $\mathrm{RE}[\eta]$ and its approximation $\mathrm{R\check{E}}[\eta]$. The middle panels do not give results for $\mathrm{MSIS}(\theta_\star)$, which uses $\mathrm{IS}(\theta_\star)$ (resp., SRS) to estimate ξ (resp., μ).

We ran simulation experiments to estimate this model's EC η for $p = 0.999$. For this model we can compute analytically the mean as $\mu = 104.02$, but this may not be possible for more complicated models, and our simulation experiments treat μ as unknown, requiring estimation. The value of ξ is not analytically tractable, and we obtained its "true" value as $\xi = 1884.6$ from an SRS simulation with sample size 10^7, giving the "true" value for EC as $\eta = 1780.6$.

We construct nominal 95% confidence intervals for η using two approaches: batching and sectioning. For an estimation method $\mathfrak{M}$ and total overall sample size n, we first construct $b \geq 2$ i.i.d. estimators $\widehat{\eta}^{(j)}_{\mathfrak{M},n/b}$, $j = 1,2,\ldots,b$, of η, each based on a sample size n/b. Batching uses their sample average $\bar{\eta}_{\mathfrak{M},b,n} = (1/n)\sum_{j=1}^{b} \widehat{\eta}^{(j)}_{\mathfrak{M},n/b}$ and sample variance $S^2_{\mathfrak{M},b,n} = (1/(b-1))\sum_{j=1}^{b}[\widehat{\eta}^{(j)}_{\mathfrak{M},n/b} - \bar{\eta}_{\mathfrak{M},b,n}]^2$ to build an approximate $\alpha =$ 0.95-level CI $I_{\mathfrak{M},b,n} = (\bar{\eta}_{\mathfrak{M},b,n} \pm t_{b-1,0.95} S_{\mathfrak{M},b,n}/\sqrt{b})$, where $t_{b-1,\alpha} = H^{-1}_{b-1}(1-\alpha/2)$, and H_{b-1} the Student-t CDF with $b-1$ degrees of freedom. Sectioning [9, Section V.5] replaces $\bar{\eta}_{\mathfrak{M},b,n}$ in $S^2_{\mathfrak{M},b,n}$ and $I_{\mathfrak{M},b,n}$ with the overall point estimator $\widehat{\eta}_{\mathfrak{M},n}$ to get a CI $J_{\mathfrak{M},b,n}$, centered at $\widehat{\eta}_{\mathfrak{M},n}$. Because η estimators are biased, with the bias shrinking (nonmonotonically) as the sample size increases, the sectioning CI can have better coverage than $I_{\mathfrak{M},b,n}$ because $J_{\mathfrak{M},b,n}$ is better centered on average ([60]).

Table 4.1 gives results of coverage experiments to construct batching and sectioning CIs for η using SRS, IS$_\rho$ (explained below), MSIS, ISDM, and DE, each with overall sample size $n = 2000$. We take $b = 10$, as suggested by [94]. From 10^3 independent replications, we estimated the batching and sectioning CIs' coverage and *average relative half width* (ARHW), and the point estimators' *root-mean-squared relative error* (RMSRE), defined as $\sqrt{E[(\hat{\eta}-\eta)^2]}/\eta$ for a generic estimator $\widehat{\eta}$ of η. When the coverage is low, the ARHW and RMSRE results may not be reliable.

For SRS, the batching CI has poor coverage, while the coverage for sectioning is reasonably close to nominal, because of the benefits of sectioning as we explained before.

Also, for sectioning, the ARHW (resp., RMSRE) for SRS is about 7 (resp., 13) times larger than for MSIS.

IS_ρ is a modification of a method of [40] for estimating a tail probability $P(Y > x)$ for a given large threshold x to estimate $P(Y > \rho)$, where ρ is either η or ξ, and then use the generated IS data to compute an estimator of η. But as these choices for ρ are unknown, we cannot directly apply the [40] IS algorithm to estimate $P(Y > \rho)$. Rather, when $\rho = \xi$, we first run $j_0 = 5$ pilot IS simulations, each with small sample size $n_0 = 100$, to estimate $P(Y > x)$ at j_0 different thresholds x, and then interpolate to obtain a crude approximation $\mathring{\xi}$ to ξ. Then IS_ξ runs another IS simulation with sample size $n - j_0 n_0$ to estimate $P(Y > \mathring{\xi})$, finally employing the generated IS data to estimate both ξ and μ to obtain an estimator of η. Each independent replication repeated these steps. The full details of this approach for $\rho = \xi$ appear in Appendix I. For ISDM, the only difference from IS_ξ is that we sample $\mathbf{X} \sim \widetilde{G}_{\text{ISDM}}$ in Section 4.4.2, where G^* corresponds to IS_ξ.

For IS_ρ with $\rho = \eta$, we execute an additional pilot SRS simulation with sample size n_0 to produce an approximation μ_* to μ, and compute $\eta_* = \mathring{\xi} - \mu_*$ as an approximation to η. Then IS_ρ for $\rho = \eta$ runs an IS simulation with sample size $n - (j_0 + 1)n_0$ to estimate $P(Y > \rho)$ for $\rho = \eta_*$, and employs the resulting IS data to compute estimators of both ξ and μ, resulting in our final estimator of η.

Table 4.1 shows that the coverage for each IS_ρ CI is at or close to 0 for all choices of ρ. This occurs because in IS_ρ, we apply the same IS data from estimating ξ to also estimate μ, leading to the problems discussed in Section 4.4 and the poor coverage for our CIs. In particular, the average across 10^3 replications of the IS_η estimator of μ is about 11.1, quite far from the true value 104.02. As noted on pp. 134–135 of [9], these types of discrepancies can occur with IS when the sample size is not sufficiently large, especially when an IS approach is applied inappropriately for a given estimation problem. (To investigate this further, we ran additional simulations (not reported) verifying that the IS_ρ CIs approach nominal coverage with larger sample size n for $p = 0.95$. For larger p,

rather than conducting converge experiments, which would require a massive sample size and extremely long CPU time (several months), we did experiments showing that the IS_ρ estimators of η appear to converge to its "true" values as n gets larger, indicating that the CIs should approach nominal coverage with a large enough sample size.) Also, the ARHW and RMSRE results for IS_ρ may not be reliable because of the poor coverages.

For MSIS, we use a total sample of size $j_0 n_0$ for computing the crude quantile approximation $\mathring{\xi}$, as is done with IS_ξ; then generate an IS sample of size $\delta(n - j_0 n_0)$ to estimate $P(Y > \mathring{\xi})$, and use the resulting IS data to estimate ξ; and finally employ an SRS sample of size $(1-\delta)(n - j_0 n_0)$ for the estimation of μ. Table 4.1 shows that MSIS sectioning and batching CIs achieve nominal coverage, with about the same ARHW, but with the sectioning point estimator having roughly 10% smaller RMSRE. MSIS outperforms SRS for both batching and sectioning, with the mean-squared error (MSE) for sectioning being reduced by a factor of $(2.276\text{e–}01/1.801\text{e–}02)^2 \approx 160$. In our python implementations, the IS code, including the pilot runs to obtain the crude quantile approximation $\mathring{\xi}$, requires about thrice the CPU time as SRS to execute. Taking this into account, MSIS improves work-normalized MSE by about a factor of 50 compared to SRS. DE and MSIS differ only in computing their estimator of η from the generated data; see (4.25) and (4.21).

We next compare the methods (MSIS, ISDM, DE of Section 4.4) that combine SRS and IS. For the i.i.d. sum model in Section 4.5, recall that Theorem 8 and Figure 4.1 (Section 4.6.1) established the following properties for the methods' (approximate and exact) RE of η:

- MSIS does better than ISDM (but not by a lot);
- Both MSIS and ISDM greatly outperform DE; and
- DE behaves about the same as the worse of IS_ξ and SRS.

For the more complicated PCRM, Table 4.1 shows that the methods perform similarly in terms of ARHW and RMSRE. First, by comparing MSIS and ISDM, we can see that MSIS has about 30% smaller ARHW and RMSRE than ISDM for sectioning; while MSIS and ISDM produce CIs that have close to nominal coverage, MSIS perhaps does a bit better. Second, relative to MSIS, DE has about 5 (resp., 10) times larger ARHW (resp., RMSE) for sectioning. While DE has sectioning coverage for η reasonably close to nominal, it is not as good as MSIS. We expect DE to do about the same as the worse of SRS and IS_ξ, and the ARHW and RMSRE of DE are reasonably close to those of SRS for sectioning, but IS_ξ has very poor coverages so its ARHW and RMSRE results may not be reliable. The coverages of batching and sectioning for DE differ substantially, which is similar to what we see for SRS, and the reason for this difference is just like that of SRS as explained before. Both ISDM and DE incur about the same CPU time as MSIS. Thus, the methods exhibit the same behavior for the PCRM as we saw for the i.i.d. sum model. Also compared to SRS, MSIS improved precision by reducing the ARHW of the sectioning CI from roughly 0.3 to only 0.04.

Table 4.1 Results Comparision

	Batching			Sectioning		
Method	Coverage	ARHW	RMSRE	Coverage	ARHW	RMSRE
MSIS	0.921	0.038	2.016e–02	0.956	0.041	1.801e–02
ISDM	0.867	0.060	5.199e–02	0.915	0.060	2.574e–02
DE	0.076	0.136*	2.274e–01*	0.884	0.220	1.803e–01
SRS	0.365	0.273*	2.633e–01*	0.892	0.292	2.276e–01
IS_η	0.096	0.024*	7.253e–02*	0.087	0.024*	7.370e–02*
IS_ξ	0.074	0.027*	5.212e–02*	0.047	0.028*	5.356e–02*

Note: We ran 10^3 independent replications of the PCRM to estimate the coverage and ARHW of sectioning and batching CIs with nominal 95% confidence level for the EC η for $p = 0.999$ estimated with sample size $n = 2000$. Numbers marked with $*$ may not reliable due to very low coverage

CHAPTER 5

HOLISTIC RESOURCE ALLOCATION UNDER FEDERATED SCHEDULING FOR PARALLEL REAL-TIME TASKS

In this chapter, we present a holistic resource allocation framework for parallel real-time tasks under federated scheduling. Under our proposed framework, in addition to dedicated cores, each parallel task is also assigned with dedicated cache and memory bandwidth resources. We study the characteristics of parallel tasks upon different resource allocations following a measurement- based approach and proposes a technique to handle the challenge of tremendous profiling for all resource allocation combinations under this approach. Further, we propose a holistic resource allocation algorithm that well balances the allocation between different resources to achieve good schedulability. Additionally, we provide a full implementation of our framework by extending the federated scheduling system with Intel's Cache Allocation Technology and MemGuard. Finally, we demonstrate the practicality of our proposed framework via extensive numerical evaluations and empirical experiments using real benchmark programs.

5.1 Related Work

Parallel real-time scheduling. The problem of scheduling parallel real-time tasks has been broadly studied. The earlier works develop a task decomposition technique to apply the analysis of multiprocessor scheduling [69, 113, 65, 133, 97, 55]. For directly scheduling parallel tasks, classic schedulers [6, 19, 81, 22, 92, 98] and Federated Scheduling that is specifically designed for parallel tasks [82, 54, 127, 3] have been analyzed. All of them, except for [3, 123, 121, 98], only consider how to allocate cores to parallel tasks and do not consider the contention in cache and memory bandwidth. Alhammad and Pellizzoni, for the first time, analyze the memory bandwidth allocation for parallel tasks, using a

theoretical approach. They model the memory time as part of the work and calculate a task's execution time given a certain number of cores and a certain amount of bandwidth assigned to the task. For analyzing private cache, a private-cache-aware algorithm is proposed for finding partitioned non-preemptive schedules for parallel tasks [98]. In contrast, we consider both shared cache and memory bandwidth for parallel tasks and takes an empirical approach based on measurements of WCET. E-WarP [121] is a framework that analyzes the fine-grained memory demand of applications and uses the developed memory enveloping to perform accurate WCET predictions under bandwidth regulation for both CPU and accelerator workload. To analyze the fine-grained cache behavior of parallel tasks, [123] incorporates the cache-aware BUNDLE-scheduling into federated scheduling for parallel tasks. Both works focus on analyzing the fine-grained behaviors of individual parallel tasks and improve their execution efficiencies. Thus, they are orthogonal to this work and can be integrated for better performance.

Allocating cache and memory bandwidth. Cache partitioning techniques, such as page coloring, have been studied extensively to reduce contention on cache [142, 35, 64, 11] (see [43] for a survey). Interference due to cache has also been incorporated into scheduler design and analysis [44, 20, 114, 137, 135]. Recent processors provide more efficient hardware support for cache partitioning [52, 7]. Analyses on memory controllers achieve deterministic memory access latency via detailed assumptions and/or modifications to controller hardware [63, 46, 84, 143, 45]. In contrast, software-based techniques regulate the memory bandwidth via throttling a core when the monitoring unit observes the excessive memory requests of the core [140, 138, 1]. CaM [136] proposes to holistically allocate cache and memory bandwidth to sequential tasks on multicore machines. This idea is later incorporated into the compositional analysis for real-time virtualization to provide better timing isolation among tasks in VMs. To address shared cache and memory bus contention while ensuring task timing requirements in virtualized systems, Maracas [137] adopts page coloring techniques and a latency-based memory throttling approach. All the

above research considers sequential tasks, while this work extends CaM for parallel tasks with federated scheduling.

5.2 Impact of Resource Allocations on Parallel Real-Time Tasks

To investigate the characteristics of the worst-case execution times of parallel real-time tasks when allocated with different amounts of cores, cache partitions, and memory bandwidth resources, in this section, we conduct an empirical evaluation using real-world parallel applications. Specifically, we extend parallel benchmark programs written in the widely used OpenMP [101] language using Intel CAT [52] and MemGuard [140] for dedicating resources in our experiments. The observations obtained from this empirical study not only motivate the importance of holistic resource allocation for parallel real-time tasks, but also stimulates us to apply a regression function on the measurement results. This regression function is later used to reduce the tremendous profiling effort for all different combinations of resource allocation in the measurement-based approach.

5.2.1 Experimental Setup

We first describe the resource allocation implementation and experimental setup for measuring the worst-case execution times of parallel benchmark programs upon different numbers of allocated cores, cache partitions, and memory bandwidth partitions.

CAT. Intel's Cache Allocation Technology (CAT) [52], which is available to Intel processors starting with the Xeon E5 v4 family, provides software-programmable control over the amount of last-level cache (LLC) that can be consumed by software or hardware threads. More specifically, CAT relies on mapping each running software or hardware thread onto an intermediate construct called a *Class of Service (CLOS)*. Then, CLOS can be configured via the L3 capacity bitmasks to set the available cache partitions, which associates the cache partitions with the software or hardware threads. Intel Resource Director Technology Software Package provides the OS interface leveraging

Linux kernel extensions to achieve the assignment of cache partitions to a process (i.e., task) or a set of cores. In our system implementation, we configure the Linux kernel via `CONFIG_INTEL_RDT_A` to enable the two OS interfaces `pqos_l3ca_set` and `pqos_alloc_assoc_set` for allocating cache partitions. For a parallel task executed on multiple dedicated cores, we use these interfaces to assign cache partitions that are shared only by the cores of the task.

Memguard. Our implementation leverages the reservation mechanism of MemGuard [140] to allocate memory bandwidth to parallel tasks and cores. Specifically, MemGuard utilizes the hardware performance monitoring counter via the Linux `perf_event` infrastructure to monitor the last-level cache miss of each core. Since each cache miss generates a memory access request, one can calculate the maximumly allowed number of cache misses for a duration without exceeding the specified memory bandwidth. When reaching this number, MemGuard throttles the computation of this core by calling the `cpu_relax()`. At the end of the current duration, MemGuard resets the counter and wakes up the core for execution. In this way, MemGuard is able to restrict the amount of memory bandwidth used by each core. However, because the hardware counter can only monitor the cache misses for each core, MemGuard only supports individually allocating a certain amount of memory bandwidth to a core. But it does not allow allocating memory bandwidth that can be shared by a set of cores or by the multiple parallel threads a process on different cores. Hence, for a parallel task assigned with multiple dedicated cores, our implementation calculates the amount of memory bandwidth to be allocated for each of these cores by using the number of cores to divide the desired total amount of memory bandwidth allocated to this task. Then, we use the interface provided by MemGuard to achieve this allocation.

Hardware. We conduct the experiments on a 14-core machine with an Intel Xeon Gold 5117 processor that supports Intel CAT. The cores in the processor share a 19.25MB L3 cache and 6-channel 32GB DDR4 DRAM with a maximum memory speed of 2400MHz. The shared L3 cache can be divided into 11 equal-size partitions. The processor has 8

Class of Service (CLOS) registers, so it supports at most 8 sets of cache partitions, where each set (i.e., each CLOS) must be assigned with at least one cache partition. We adopted the DRAM controller saturation analysis in [121] for obtaining the maximum memory bandwidth. For the workload with stores that always result in DRAM row misses, the maximum memory bandwidth without fully saturating the DRAM controller is 7.83 GB/s. In comparison, for the read-intensive workload that always results in cache misses, the maximum memory bandwidth without fully saturating the DRAM controller is 17.97 GB/s. We observe that the realistic benchmark programs described below typically generate more loads than stores. Hence, we consider a maximum guaranteed memory bandwidth of 12GB/s assuming a ratio of roughly two stores and one load. To discretize the amount of memory bandwidth that can be allocated to tasks, we divide the bandwidth into 20 partitions of 600MB/s each, where each task is assigned with one or multiple partitions. This number is chosen considering the balance between the sufficient number of partitions for the allocation and the sufficient size of each partition.

System configuration. Our experiments are run on Linux 4.15.0, where hyper-threading, SpeedStep, and hardware cache prefetcher features are disabled to reduce the non-determinism in the timing behavior of tasks. For both the benchmark profiling described here and the empirical evaluation of our framework in Section 5.7, we run the benchmark programs under the Linux real-time priorities. Note that Linux comes with a safeguard mechanism that throttles the execution under real-time priorities when reaching 95% CPU utilization by default. We disable the real-time scheduler throttling by setting `sched_rt_runtime_us` to -1. We further reserve one core (i.e., core 0) for system services, dedicated one cache partition to this core using CAT, and restrict its memory bandwidth usage using Memguard to limit the interference from system services to the experiments. In summary, we leave 13 cores, 10 cache partitions, and 20 memory bandwidth partitions for running experiments.

Parallel runtime system. We use GCC 7.4.0 with OpenMP 2.0 as the compiler and runtime system for executing parallel benchmarks. We configure OpenMP to generate and pin exactly one thread per core using `omp_set_num_threads()` and `pthread_setaffinity_np()`, so each parallel task uses and only uses its dedicated cores without thread migrations. To further reduce the variation of parallel execution times, we set `OMP_WAIT_POLICY` as active, disallow nested parallelism using `omp_set_nested(0)`, and set `GOMP_SPINCOUNT` as infinity.

Workload. We modified 12 parallel benchmark programs to enable the allocation of dedicated cores, cache partitions, and memory bandwidth partitions using the aforementioned interfaces provided by OpenMP, Intel CAT, and MemGuard, respectively. The 12 benchmark programs are converted from two widely used parallel benchmark suites that collect real-world applications with various parallel structures and properties. Specifically, `Facesim`, `Bodytrack`, `Fluid Animate` (`Fluid`), `Swaptions`, and `Blackscholes` are from the Princeton Application Repository for Shared-Memory Computers (PARSEC) benchmark suites [16]; While `Ray Casting` (`RayCast`), `Breadth First Search` (`BFS`), `Comparison Sort` (`Sort`), `Dictionary`, `Minimum Spanning Forest` (`MSF`), `Remove Duplicates` (`RemDup`), `Nbody` are from the CMU Problem Based Benchmark Suite (PBBS benchmark suite) [111]. Note that the PBBS benchmark programs were originally written using Cilk Plus [51] and are converted into OpenMP. These benchmark programs cover a broad range of real-world applications, such as computational biology, graphics, basic building blocks, finance, computer vision, and physics simulation algorithms. They also include different representative parallel structures. For instance, `Blackscholes` performs financial analysis and is parallelized by spawning and synchronizing OpenMP tasks; `Bodytrack` is a computer vision application, which is parallelized with nested parallel for loops and has ample parallelism; In contrast, `Nbody` is a scientific application and has a more complex parallel structure with both parallel for loops and spawning tasks. The

different parallel structures not only affect the speedup of the benchmark programs, but also have impacts on their sensitivities to different allocated resources.

5.2.2 Impact of Core, Cache and Memory Bandwidth Allocations

The goal of this empirical study is to examine how the timing behavior of real-world parallel applications changes, when they are assigned with different numbers of dedicated cores, cache partitions, and memory bandwidth partitions. For brevity, we use ***MBW*** or *bandwidth* to refer to memory bandwidth in the rest of the paper.

Experiment. We run each benchmark program on increasing numbers of cores, cache partitions, and memory bandwidth partitions. Under each resource allocation, we measure the execution time of the benchmark. The profiling is conducted in a setup that mimics the execution environment of co-running multiple tasks on their dedicated resources and creates as much system-level interference as possible. Specifically, we run each benchmark under profiling at a high real-time priority. Additionally, we co-run another interfering parallel task at a lower real-time priority and allocate all the remaining resources to this task. Extending the method in [141, 132], this interfering program is essentially a parallel cache-bomb and memory-intensive program that we develop by parallelizing the Stream Benchmark [89] (similar to the benchmarks in [132]) using OpenMP. This program generates intensive memory access requests by performing read and write operations on long-vectors with minimum data re-use (either in registers or in cache). In addition, it runs in parallel on the assigned cores by OpenMP parallel for loops, which frequently synchronizes using the underlying Linux futexes. In this way, it not only compete with the benchmark under profiling on the shared DRAM controller and MSHRs, but also tries to generate some contention over the internal kernel data structures related to futexes.

Ideally, one would run each benchmark program hundreds or thousands of times to measure the ***worst-case execution time*** (***WCET***) of a benchmark. However, with 13 cores, 10 cache partitions, and 20 memory bandwidth partitions, there are a total of 2,600 distinct

combinations of resource allocations. Moreover, some benchmark programs take tens of seconds for one run. Hence, the measurement of one benchmark for the 2,600 combinations takes up to half a day, even when we run it once for each resource allocation. This circumstance motivates us to develop a regression analysis in Section 5.2.3, which enables using a much smaller number of measurements to guide the initial resource allocation for tasks. In this study, our focus is on the variety of benchmark applications and the trend of execution times upon different resource allocations, instead of obtaining the safe WCET values. Hence, we only measure the execution time of each benchmark under each of the 2,600 combinations once.

Results. Figure 5.1 shows the measurement results of four representative benchmarks. In particular, `RayCast` is a graphics rendering algorithm that uses the geometric algorithm of ray tracing to render and calculate the first intersecting triangles for rays that penetrate a 3-dimension bounding box containing a set of triangles. `BFS` performs a breadth-first search on a graph with a reasonably large size. `Blackscholes` calculates the prices for options using the Black-Scholes partial differential equation, which involves expensive computation on relatively small data. `Nbody` calculates the motion of particles under the influence of mutual gravitational forces in a dynamic system.

We calculate the speedup of a benchmark program upon a particular resource allocation. The ***speedup*** is defined as the ratio between the execution time measured for this resource allocation and the execution time when assigning only one core, one cache partition, and one memory bandwidth partition, the latter of which is also the maximum execution time of this benchmark. For better readability, we plot Figure 5.1 in terms of *slowdown*, which is the inverse of speedup. The comparison between these three representative benchmarks reveals the following observation.

Observation 1 The impact of the core, cache partition, and memory bandwidth allocations varies across different parallel benchmark applications.

Not surprisingly, the timing behaviors of different applications vary, since they have different characteristics. For example, Figures 5.1(a), 5.1(d), and 5.1(g) show the speedup of the three benchmarks on increasing numbers of cores and cache partitions, while the number of memory bandwidth partitions is fixed to one. By comparing them, we can see that cache barely makes any effect on the execution times of `Blackschole` and has a slightly larger impact on `BFS`, while it significantly affects the execution times of `RayCast`. A similar trend can be seen for memory bandwidth partitions in Figures 5.1(b), 5.1(e), and 5.1(h). When memory bandwidth allocation increases, the running times of `RayCast` and `BFS` decrease dramatically. In contrast, memory bandwidth has little impact on `Blackschole`, especially when it is running on more than 2 cores. This is because both `RayCast` and `BFS` perform computation on large data, while the intensive computation of `Blackscholes` is performed on much smaller data.

In general, well-written parallel tasks are sensitive to cores in most cases. In fact, in many cases, the execution times decrease the fastest when increasing the number of allocated cores. But the specific speedup achieved by an application when running on multiple cores depends on the parallelism of the application. For example, `Blackschole` has ample parallelism and is able to achieve near-linear speedup. In comparison, `Nbody` only has about a 25% reduction in execution times when assigned with 13 cores.

Depending on the characteristics of applications, some (e.g., `RayCast`) are sensitive to both cache and memory bandwidth resources, and some are only sensitive to memory bandwidth (e.g., `BFS`), while the others are not sensitive to cache nor memory bandwidth (e.g., `Blackschole` and `Nbody`). Somewhat surprisingly, from the benchmark applications that we profiled, we do not find any benchmark that is more sensitive to cache than to memory bandwidth. We suspect that this is both related to the memory footprint and the memory access pattern of an application.

Based on the above findings, we classify all the 12 benchmark applications into 3 large categories: cache- and MBW-sensitive benchmarks, MBW-sensitive benchmarks,

and cache- and MBW-insensitive benchmarks. As shown in Table 5.1, out of the 12 benchmark programs, there are 2 cache- and MBW-sensitive benchmarks, 6 MBW-sensitive benchmarks, and 6 cache- and MBW-insensitive benchmarks. Note that there is a continuous spectrum from being sensitive to both cache and memory bandwidth to being sensitive to memory bandwidth only. Even for those benchmarks that are considered MBW-sensitive, increasing the number of allocation cache partitions can still reduce its execution times, albeit very slightly.

Observation 2 The impacts of cache and memory bandwidth partitions on the execution times of a parallel benchmark are correlated.

Not surprisingly, this observation is also similar to what was observed for sequential tasks [136]. Figure 5.2 presents the speedup of `Facesim` under different resource allocations. We can see that increasing the number of cache partitions reduces the execution times of `Facesim` more when given 1 memory bandwidth partition; whereas increasing the number of cache partitions reduces its execution times extremely slightly when given 10 or 20 memory bandwidth partitions. This is because when a task receives little memory bandwidth, it can be throttled frequently due to running out of bandwidth reservation. Increasing the cache size can reduce the frequency of memory accesses, and thus reduce the frequency of being throttled. In contrast, the time spent on computing dominates the overall execution time, when the memory bandwidth is abundant. We observe similar behavior for `RayCast`, `Sort`, and `Dictionary`.

Observation 3 For a particular cache and memory bandwidth allocation, the execution time $\widehat{e}(th_i)$ of a benchmark on th_i dedicated cores follows the formula below:

$$\widehat{e}(th_i) = f_\infty + \frac{f_1 - f_\infty}{(th_i)^c} \tag{5.1}$$

where c is some constant, f_1 represents the total work, and f_∞ represents the span (i.e., the execution time on an infinite number of cores) upon the particular cache and memory bandwidth allocation.

Figure 5.3 shows how the execution times of `RemDup`, `Sort` and `Fluid` change on increasing numbers of dedicated cores. In addition, we apply nonlinear regression using a function in the form of Equation (5.1) to fit the profiling results. The design of the function is inspired by the theoretical analysis of the running time $e(p)$ of a parallel program when executed by a work-conserving (i.e., greedy) scheduler on increasing numbers of cores, which is essentially following the Amdahl's Law [4]. In this analysis [82], the effects of cache and memory bandwidth are ignored, and $e_p = e_\infty + \frac{e_1 - e_\infty}{p}$, where e_1 is the total work, e_∞ is the span, and p is the number of allocated cores. Note that this classical result is almost identical to our designed function, except that the constant c in Equation (5.1) is always 1 under the theoretical analysis.

From Figure 5.3, we can first observe that `RemDup` and `Sort` are memory bandwidth sensitive benchmarks, where the execution times decrease significantly given more memory bandwidth partitions. `Fluid`, on the other hand, is insensitive to cache and memory bandwidth. Moreover, the designed function can accurately approximate the trend of the measurement results for `Sort` and the obtained constant c is 1. In contrast, to obtain a low error in the regression, the constant c is 0.9 for `Fluid` and 1.9 for `RemDup`. Unlike the classical analysis that assumes linear speedup of the parallel region, our profiling results indicate that the speedup can be superlinear or sublinear. Hence, in our nonlinear regression, we do not restrict c to 1. We also notice that allowing different values of c for the same benchmark upon different cache and memory bandwidth can slightly improve the accuracy of the regression. But the variation of c is relatively small. Hence, to reduce the number of variables in the regression, we decide to use a single constant c for the same benchmark.

Furthermore, we can see that the obtained values for f_1 and f_∞ for the same benchmark program vary a lot when different numbers of cache and memory bandwidth partitions are allocated to this benchmark. Intuitively, the cache and memory bandwidth

allocation affects the latency of obtaining the data for computing, which adds to the computation time of the benchmark.

5.2.3 Fitting WCET using Nonlinear Regression

As discussed in Section 5.2.2, the enormous number of resource allocation combinations causes the time to profile the worst-case execution time (WCET) of a benchmark program tremendously long, if the profiling must be done for each of the combinations. This fact motivates us to investigate applying nonlinear regression analysis to fit the measurement results of benchmarks. Our goal here is two-fold: (1) we want to see how accurate the nonlinear regression can be when a reasonable function is used; and (2) we want to see whether it is possible to perform the measurement only for a small number of combinations, apply the non-linear regression, and obtain relatively accurate estimations on the execution times.

To answer the first question, we design the following function based on Observation 3 above for estimating the (worst-case) execution times $\widehat{e}(th_i, cp_i, mp_i)$ of a benchmark program when it is assigned with th_i cores, cp_i cache partitions, and mp_i memory bandwidth partitions:

$$\widehat{e}(th_i, cp_i, mp_i) = f_\infty(cp_i, mp_i) + \frac{f_1(cp_i, mp_i) - f_\infty(cp_i, mp_i)}{th_i^{c_{00}}} \tag{5.2}$$

where c_{00} is a coefficient. Similar to Equation (5.1), $f_1(cp_i, mp_i)$ and $f_\infty(cp_i, mp_i)$ represent the work and span of the benchmark upon cp_i cache and mp_i memory bandwidth partitions. And the coefficient c_{00} corresponds to the parameter c in Equation (5.1), which can be smaller, equal to, or larger than 1, as discussed under Observation 3.

The design of functions $f_1(cp_i, mp_i)$ and $f_\infty(cp_i, mp_i)$ is inspired by Observation 2. They try to capture the individual effect of cache or memory bandwidth allocations, as well

as the correlation between them. Specifically, they are in the following forms:

$$f_1(cp_i, mp_i) = c_{11} * (cp_i + c_{12})^{-c_{17}} * (mp_i + c_{13})^{-c_{18}}$$
$$+c_{14} * (cp_i + c_{12})^{-c_{17}} + c_{15} * (mp_i + c_{13})^{-c_{18}} + c_{16} \quad (5.3)$$
$$f_\infty(cp_i, mp_i) = c_{21} * (cp_i + c_{22})^{-c_{27}} * (mp_i + c_{23})^{-c_{28}}$$
$$+c_{24} * (cp_i + c_{22})^{-c_{27}} + c_{25} * (mp_i + c_{23})^{-c_{28}} + c_{26} \quad (5.4)$$

where c_{11} to c_{28} are also coefficients.

Table 5.1 Mean Relative Error of Fitting WCET

Cache- and MBW-sensitive benchmarks:

Benchmark	RayCast	Facesim
MRE	0.05283	0.04706

MBW-sensitive benchmarks:

Benchmark	BFS	Sort	Dictionary	MSF	RemDup
MRE	0.08200	0.03881	0.09334	0.0526	0.06961

Cache- and MBW-insensitive benchmarks:

Benchmark	Bodytrack	Blackscholes	Fluidanimate	Nbody	Swaption
MRE	0.03749	0.05648	0.03948	0.00272	0.07791

Results. We use the nonlinear regression tool of `curve_fit` in Python `scipy.optimize` library to fit the measured execution times of benchmark programs. Other nonlinear regression tools, such as Matlab Cftool, DataFit from Oakdale Engineering, Origin from OriginLab, and 1stOpt from 7D-Soft, can also be used. We do not observe any difference in the results in terms of relative errors when using different tools. We initialize all the coefficients (i.e., c_{00} to c_{28}) to 1, since this initialization often leads to faster convergence in practice. One could also initialize the coefficients to any other random values. As long as the regression converges and the relative error is small, the values of the coefficients

are similar. To evaluate the performance of the nonlinear regression, we calculate the relative error of the approximated execution times from the measured execution times for each allocation setting and report the ***mean relative error*** (***MRE***). Formally, mean relative error is calculated as

$$MRE = \frac{1}{n}\sum_{1}^{n}\frac{|\widetilde{e}(th_i, cp_i, mp_i) - \widehat{e}(th_i, cp_i, mp_i)|}{\widehat{e}(th_i, cp_i, mp_i)}$$

Table 5.1 summarizes the mean relative errors of the nonlinear regression results for different benchmark programs. Results show that the mean relative errors of the fitted execution times are smaller than 10% for all benchmarks. The accuracy of the regression does not seem to be correlated to the type of benchmark programs. Note that although our experiments apply the nonlinear regression to the measurement of execution time for one run under each allocation setting, in principle, this approach is applicable to the WCET measurements of multiple runs.

Regression using a smaller number of sampled data points. Here, we explore whether using only a small number of measurements suffices to achieve comparable accuracy with using the measurement results of all resource allocation settings. In particular, we start with feeding the regression with only 125 sampled data points. The samples come from the execution times when assigning [1, 3, 5, 7, 9] cache partitions, [1, 5, 10, 15, 20] memory bandwidth partitions, and [1, 4, 7, 10, 13] cores. The 125 initial samples are specifically chosen to evenly span the entire space. Next, we randomly sample 75 data points and add them into the regression, with the hope that more data can improve the accuracy. We repeat this process until all the 2,600 data points have been added to the regression.

Figure 5.4 presents the mean relative errors for 6 representative benchmark programs when increasing the number of sampled data points used for the nonlinear regression. Note that in practice, one can only calculate the mean relative error using the sampled data points. Hence, in addition to reporting the mean relative errors calculated using all the 2,600 data

points (i.e., global data), we also report the mean relative errors calculated using only the sampled data points.

First and foremost, we can see that, for all benchmarks, using only about 250 data points the nonlinear regression can already achieve comparable performance to using all the data points. Therefore, our designed regression function gives the potential of significantly fewer measurements for soft real-time systems. For hard real-time systems, after obtaining the candidate resource allocation for a task set, one can conduct extensive profiling of WCET upon the allocated resources. If the WCET ends up exceeding the deadline, then local refinement and profiling can be performed to adjust the resource allocation till all tasks can meet their deadlines.

We also observe that the trends of the MRE calculated using the global data and using the sample data are not necessarily similar. This leads to a natural question of when to stop sampling more data points (i.e., conducting more measurements). For benchmarks like `Nbody`, the initial 125 data points already achieve a very low error, which means that the regression function can very nicely approximate the true data. For the other benchmark programs, sampling another small amount of data points as a validation set and using it to determine when to stop can be a good choice.

5.3 Problem Specification and Prior Results

The empirical study of benchmarks motivates the importance of holistic resource allocation for parallel real-time tasks. This section presents the formal model for this scheduling problem based on the timing characteristics of tasks observed in our measurements.

System model. We consider a machine with N_{th} cores, sharing an L3 cache with N_{cp} equal partitions and a memory bus with N_{mp} equal memory bandwidth partitions. We extend the federated scheduling [82] introduced in Section 5.1 for scheduling parallel real-time tasks to incorporate cache and MBW allocation. In particular, federated scheduling forces all low-utilization tasks to run sequentially. We assume that these tasks are scheduled either by

the partitioned *earliest deadline first* (*EDG*) algorithm or by the partitioned *rate monotonic* (*RM*) algorithm on its partitioned core. Each of the remaining high-utilization tasks is allocated with some dedicated cores, where it runs in parallel. The resource allocation is done via the class of service (CLOS). A CLOS can either be associated with one parallel task with a set of dedicated cores, or with one core with sequential tasks partitioned to it. Additionally, a CLOS is assigned to a set of dedicated cache partitions and a number of dedicated MBW partitions. The minimum number of cache and MBW partitions assigned to a CLOS is one.

Task model. We seek to schedule a set of m tasks. Each task τ_k is modeled as a 3-tuple $\tau_k = \{e_k(th_i, cp_i, mp_i), p_k, d_k\}$, where p_k is its period, d_k is its deadline, and $e_k(th_i, cp_i, mp_i)$ is the measured WCET of the task when executed alone on th_i cores with cp_i cache and mp_i MBW partitions. In this work, we focus on tasks with implicit deadlines where $p_k = d_k$. Similar to [136], we define $re_k = e_k(1, N_{cp}, N_{mp})$ as the ***reference WCET*** and calculate the ***reference utilization*** as $ru_k = e_k(1, N_{cp}, N_{mp})/p_k$. In addition, we also denote $pe_k = e_k(1,1,1)$ as the ***peak WCET*** of τ_k and calculate the ***peak utilization*** as $pu_k = e_k(1,1,1)/p_k$. Thus, the ***speedup*** of a task under a certain resource allocation is $t\_speedup_k(th_i, cp_i, mp_i) = pe_k/e_k$. A task is schedulable if it can always finish its execution before its deadline, and a task set is schedulable if all tasks are schedulable.

Problem/Objective. For a multicore machine with a shared L3 cache and memory bus, our goal is to develop a strategy for allocating resources, including cores, cache, and memory bandwidth partitions, to parallel real-time tasks, so that the task set is schedulable.

Most relevant work. The theoretical modeling, as well as the allocation strategy for low-utilization tasks on the partitioned cores, follows the CaM proposed by Xu et al. [136]. Here, we briefly introduce the high-level strategy of ***CaM***, which is slightly modified and used as the subroutine of our proposed algorithm in Section 5.5. It first uses a clustering algorithm to group the tasks by their sensitivity. It then tries to put tasks of the same group onto the same core, while maintaining the reference utilizations of cores roughly the same.

Cache and MBW partitions are assigned to the over-utilized cores that have the maximum decrease in their total utilizations of the partitioned tasks. Finally, it iteratively moves tasks from over-utilized cores to under-utilized ones and re-assigns cache and MBW partitions, until all cores become schedulable or it exceeds the maximum allowed iterations.

The main difference between the models in this work and in [136] is that tasks are parallel and may need to run on multiple cores to meet their deadlines. To handle parallel tasks, we proposed to extend federated scheduling by holistically allocating cache and MBW resources. The original ***federated scheduling*** assigns dedicated cores to parallel tasks with ***high-utilizations*** (i.e., utilization larger than one), forces the remaining ***low-utilization*** tasks to run sequentially, and partition these sequential tasks on the remaining cores.

Challenge. Incorporating cache and MBW resources for federated scheduling introduces several challenges: (1) how to distinguish high- and low-utilization tasks when a task has different utilizations given different numbers of cache and MBW partitions; (2) how to allocate a reasonable combination of cores, cache, and MBW partitions to a high-utilization task; (3) how to reserve enough cores, cache, MBW partitions for the set of low-utilization tasks.

5.4 Optimal Algorithm.

In this section, we present our ***mixed-integer nonlinear programming*** (***MINLP***) with nonlinear constraints for this resource allocation problem. Constructing the MINLP and its corresponding constraints is not straightforward. One of the reasons is that high-utilization tasks are allocated with dedicated cores while low-utilization tasks are partitioned onto shared cores. For example, one needs to create MINLP variables to decide and distinguish whether a task is high- or low-utilization. Furthermore, variables are also needed to distinguish whether a core is shared by some low-utilization tasks or dedicated to a high-utilization task. The core shared by low-utilization tasks needs to be allocated with at

least one cache partition and one memory bandwidth partition, in order to execute tasks. In contrast, the cores that are dedicated to a high-utilization task share the cache and memory bandwidth partitions allocated to that task.

We address all the above challenges and develop the following MINLP formulation, which can be solved using the existing solver in SCIP Optimization Suite [36]. The notations used in the MINLP formulation are summarized in Table 5.2. All the variables except for the last one are non-negative integers. To illustrate the intuitions for the MINLP formulation, we use the simple task set in Figure 5.5 as an example.

At a high level, the MINLP uses the binary variable β_i to distinguish whether a task is high-utilization (needs dedicated resources) or low-utilization (partitioned to a core shared with other low-utilization tasks). For instance, in Figure 5.5 Task 2 is considered as a high-utilization task. Note that a task may change from high to utilization by increasing the allocated cache and memory partitions to reduce its execution time. Similarly, γ_j specifies whether a core is dedicated to a high-util task or shared by low-util tasks. For example, Core 3 is shared by Tasks 1 and 3. The mapping between the tasks and cores is stored in $\zeta_{i,j}$, which is highlighted in green in Figure 5.5. Depending on whether a core is shared by tasks and whether a task is assigned with multiple cores, we can distinguish the type of a core and the type of a task, respectively.

The critical reason for the need for separate indicator variables to distinguish the type of tasks and cores is that the cache and memory bandwidths associated with them are different. In particular, low-utilization tasks that are partitioned onto the same core share the cache and bandwidth partitions associated with this core (e.g., Tasks 1 and 3 on Core 3 share 4 cache and 8 bandwidth partitions). In contrast, a high-utilization task has dedicated cache and bandwidth partitions (e.g., Task 2 on Cores 2 and 4 has 1 cache and 2 bandwidth partitions), which are essentially shared by this task's dedicated cores. Hence, the constraints for cache and bandwidth partitions can only be properly formulated when the task and core types are clear. Such information is specified by $CPtask_i$ and $MPtask_i$ for

high-utilization tasks, and by $CPcore_j$ and $MPcore_j$ for the cores shared by low-utilization tasks. All these requirements are encoded into the MINLP constraints to obtain the optimal solution to the resource allocation problem.

Table 5.2 Notations

m	Number of tasks in the task set
N_{th}	Number of cores on the hardware
N_{cp}	Number of L3 cache partitions on the hardware
N_{mp}	Number of memory bandwidth partitions on the hardware
α_i	Binary: if task i has a valid resource allocation
β_i	Binary: if task i is high-util with dedicated resources; or, low-util task
γ_j	Binary: if core j is dedicated to a high-util task; or, shared by low-util tasks
$\zeta_{i,j}$	Binary: if task i executes on core j
$CPtask_i$	Number of cache partitions allocated to (high-util) task i
$MPtask_i$	Number of bandwidth partitions allocated to (high-util) task i
$CPcore_j$	Number of cache partitions allocated to (low-util) core j
$MPcore_j$	Number of bandwidth partitions allocated to (low-util) core j
$e_i(TH_i, CP_i, MP_i)$	Measured WCET of task i when executed on the specified resources

We now formally describe the constraints that our MINLP formulation encodes.

C1) This constraint essentially distinguishes whether a task has a valid resource allocation using properties of the task-core mapping $\zeta_{i,j}$. The variable α_i is only used in the optimization objective to maximize the number of schedulable tasks. When task i does not have a valid resource allocation (i.e., $\alpha_i = 0$), then no cores should have been allocated to this task. When there exists a valid allocation, the number of assigned cores should not

exceed the total available cores N_{th}.

$$\forall 1 \leq i \leq m: \quad \alpha_i \leq \sum_{j=1}^{N_{th}} \zeta_{i,j} \leq N_{th} \cdot \alpha_i$$

C2) In the task-core mapping $\zeta_{i,j}$, high-utilization tasks (i.e., $\beta_i = 1$) cannot share the same core j. Similarly, a low-utilization task i (i.e., $1 - \beta_i = 1$) cannot be assigned to more than one core.

$$\forall 1 \leq j \leq N_{th}: \quad \sum_{i=1}^{m} (\zeta_{i,j} \cdot \beta_i) \leq 1$$

$$\forall 1 \leq i \leq m: \quad \sum_{j=1}^{N_{th}} \left(\zeta_{i,j} \cdot (1 - \beta_i)\right) \leq 1$$

C3) This constraint essentially distinguishes the type of a core using properties of the task-core mapping. In the task-core mapping $\zeta_{i,j}$, if core j is dedicated (i.e., $\gamma_j = 1$), there is exactly one high-utilization task (i.e., $\beta_i = 1$) executing on this core. Otherwise, this core is shared, so no high-utilization task should execute on this core.

$$\forall 1 \leq j \leq N_{th}: \quad \sum_{i=1}^{m} (\zeta_{i,j} \cdot \beta_i) = \gamma_j$$

C4) This constraint essentially distinguishes the type of a task using properties of the task-core mapping. If task i executes on one or multiple dedicated cores (i.e., $\gamma_j = 1$), this task must be high-utilization (i.e., $\beta_i = 1$).

$$\forall 1 \leq i \leq m: \quad \sum_{j=1}^{N_{th}} \left(\zeta_{i,j} \cdot \gamma_j\right) \leq \beta_i \cdot N_{th}$$

C5-1) The next four sets of constraints restrict the allocation of cache and bandwidth partitions. First, a dedicated core (i.e., $\gamma_j = 1$) must share the partitions with the cores belonging to the same high-utilization task, so it does not have any exclusive partitions.

$$\forall 1 \leq j \leq N_{th}: \quad \gamma_j \cdot CPcore_j = 0 \quad \wedge \quad \gamma_j \cdot MPcore_j = 0$$

C5-2) Second, a shared core (i.e., $\gamma_j = 0$) must have at least one cache partition and one bandwidth partition.

$$\forall 1 \leq j \leq N_{th}: \quad \gamma_j + CPcore_j \geq 1 \quad \wedge \quad \gamma_j + CPcore_j \geq 1$$

C5-3) Next, a low-utilization task (i.e., $\beta_i = 0$) does not have any allocated cache or bandwidth partition. For high-utilization tasks, the number of allocated partitions is bounded by availability.

$$\forall 1 \leq i \leq m: \quad CPtask_i \leq \beta_i \cdot N_{cp} \quad \wedge \quad MPtask_i \leq \beta_i \cdot N_{mp}$$

C5-4) Lastly, a high-utilization task (i.e., $\beta_i = 1$) must be allocated with at least one cache partition and one bandwidth partition.

$$\forall 1 \leq i \leq m: \quad CPtask_i \geq \beta_i \quad \wedge \quad MPtask_i \geq \beta_i$$

C6) The next two constraints bound the total number of allocated cache (or bandwidth) partitions.

$$\sum_{i=1}^{m} (\beta_i \cdot CPtask_i) + \sum_{j=1}^{N_{th}} \left(\gamma_i \cdot CPcore_j\right) \leq N_{cp}$$

$$\sum_{i=1}^{m} (\beta_i \cdot MPtask_i) + \sum_{j=1}^{N_{th}} \left(\gamma_i \cdot MPcore_j\right) \leq N_{mp}$$

C7-1) The last sets of constraints make sure the task is schedulable. For a high-utilization task i that is allocated with $\sum_{j=1}^{N_{th}} \zeta_{i,j}$ cores, $CPtask_i$ cache, and $MPtask_i$ bandwidth partitions, its execution time e_i should be no more than its implicit deadline p_i. In addition, we also require that this task is allocated with the minimum number of cores, i.e., reducing one dedicated core would result in deadline misses.

$$\forall 1 \leq i \leq m: \quad \beta_i \cdot e_i \left(\sum_{j=1}^{N_{th}} \zeta_{i,j}, CPtask_i, MPtask_i \right) \leq p_i$$

$$\forall 1 \leq i \leq m: \quad \beta_i \cdot e_i \left((\sum_{j=1}^{N_{th}} \zeta_{i,j} - 1), CPtask_i, MPtask_i \right) \geq p_i$$

C7-2) For a core with $CPcore_i$ allocated cache and $MPcore_i$ bandwidth partitions shared by low-utilization tasks (i.e., $1 - \gamma_j = 1$), the total utilization of these tasks cannot exceed 1.

$$\forall 1 \leq j \leq N_{th}: \quad (1 - \gamma_j) \cdot \sum_{i=1}^{m} \left((1 - \beta_i) \cdot \zeta_{i,j} \cdot \frac{e_i(1, CPcore_j, MPcore_j)}{p_i} \right) \leq 1$$

Objective: Our goal is to maximize the number of tasks that can be feasibly scheduled. This can be formulated as follows using the indicator variable α_i:

$$maximize \sum_{i=1}^{m} \alpha_i$$

Improved MINLP implementation. While being optimal, the complexity of solving this MINLP problem is extremely high, making it very inefficient to use in practice. Moreover, the existing solvers for MINLP need to convert the MINLP to regular nonlinear programming before adding the integer constraints, so the non-continuous and nonlinear $e_k(th_i, cp_i, mp_i)$ cannot be directly used. Instead, we must use Function (5.2) obtained from fitting the measured data, which further reduces the efficiency. Thus, we further improve the implementation of the MINLP formulation.

In particular, we separate the allocation to high- and low-utilization tasks and use a brute-force method to enumerate all the good allocations for high-utilization tasks. Once the allocation decisions for the high-utilization tasks are made, the remaining problem becomes the resource allocation problem for sequential (low-utilization) tasks. For this problem, there exists a mixed-integer programming (MIP) formulation [136], which is much faster than the original MINLP formulation for the entire task set.

Thus, the key to a good improved implementation is on deciding (1) which tasks are high-utilization tasks and (2) how many resources should be allocated to high-utilization

tasks. To maintain the optimality of the MINLP formulation, all possible choices for the above two questions must be verified before determining that the task set is unschedulable. However, not all choices have equal importance — the ordering of these choices crucially affects the running time of the implementation for most task sets. This is because once a schedulable allocation decision is found, all the remaining choices no longer need to be verified.

With this intuition, we first construct all possible subsets of tasks in the original task set. If one subset is chosen as high-utilization tasks, the supplement subset becomes low-utilization tasks. Among these possible subsets, we sort them according to their total reference utilization from large to small. Thus, the tasks with higher utilizations will be considered as high-utilization tasks first. Next, for a considered subset, we verify all combinations of resource allocation to these high-utilization tasks and see if any combination can make all tasks schedulable. Here, we prune some combinations that are clearly not feasible or reasonable. For example, if a task is not schedulable given a set of resources, then it is also not schedulable given strictly fewer resources. Moreover, among all the combinations that are schedulable, it is obviously more beneficial to the remaining low-utilization tasks if strictly more resources remain. Hence, instead of running the MIP formulation of the low-utilization tasks for all schedulable combinations, we prune those that use strictly more resources. In this way, only the combinations at the Pareto boundary are verified, which significantly reduces the running time of this implementation in practice.

5.5 Holistic Resource Allocation for Federated Scheduling

Although the improved MINLP implementation reduces the running time by a lot, it can still take a long time to obtain the optimal results, especially when the problem size is large. Therefore, we propose a heuristic-based strategy that has comparable or slightly worse performance with a running time in orders of magnitude shorter than the MINLP

formulation. This section first gives an overview of our holistic resource allocation strategy and then provides details of the algorithm.

5.5.1 Algorithm Overview

The holistic resource allocation algorithm for parallel real-time tasks leverages the advantages of CaM [136] and federated scheduling [82], as well as insights obtained from our empirical study using real-world benchmarks in Section 5.2. In particular, our holistic algorithm is based on the following high-level strategies.

1. We assign dedicated cores, cache partitions, and memory bandwidth partitions to each high-utilization task;

2. We use the Speedup profile $t\_speedup_k(th_i, cp_i, mp_i)$ to cluster low-utilization tasks to try to partition tasks with similar sensitivity onto the same core, while also maintaining relatively balanced workload between low-utilization cores;

3. We use a dynamical threshold for distinguishing high-utilization and low-utilization tasks;

4. We assign each available resource instance to the unschedulable task or core that receives the largest positive benefit (i.e., the largest reduction in its utilization);

5. We allow an unschedulable core with low-utilization tasks to exchange resources with a schedulable high-utilization task, under the condition that the high-utilization task remains schedulable and the low-utilization core has a reduction in its total utilization after the exchange.

Algorithm 2 gives the main steps of our holistic resource allocation algorithm, which composes of five phases:

(1) Phase 1 (Lines 1–3) dynamically classifies all tasks into high- and low-utilization tasks, by iteratively moving more tasks from low-utilization to high-utilization set. The set of high-utilization tasks τ^{high} is initialized as the tasks with high reference utilizations $ru_k = e_k(1, N_{cp}, N_{mp})/p_i \geq 1$. At each iteration, one more low-utilization task with the largest reference utilization is selected by the *selectAddHT()* procedure to be moved from τ^{low} to τ^{high}.

(2) Phase 2 (Lines 4–7) assigns cores, cache and MBW partitions to each high-utilization task τ_k in τ^{high} until it can meet its deadline (i.e., $e_k(th_i, cp_i, mp_i) \leq d_k$). The allocation follows the largest benefit first, to be explained in Section 5.5.2. During the assignment, any task in τ^{low} that was considered as low-utilization tasks but can no longer meet its deadline by running sequentially on the remaining resources for low-utilization tasks is moved to τ^{high} and participate the resource assignment process. If the available resources are not even enough for high-utilization tasks, the system is deemed unschedulable. Otherwise, the procedure *initLR()* calculates the unallocated resources to be used for partitioning low-utilization tasks.

(3) Phase 3 (Lines 8–13) tries to cluster the low-utilization tasks based on their speedup profile, partition them on the remaining cores, and assign cache and MBW partitions to each core using the procedure *CaMAllocLR*. It is almost the same as the heuristic resource allocation algorithm in CaM [136], except for some differences in its balance procedure. In this procedure, a low-utilization task can be migrated from one core to another. Here, we need to make sure that after the migration, this task still remains as a low-utilization task and its original core does not become empty. If all low-utilization tasks are schedulable on their partitioned cores, the task set is schedulable.

(4) Phase 4 (Lines 14–15) is only reached if there is some core with unschedulable low-utilization tasks. In this case, via procedure *resEx()* the unschedulable low-utilization cores try to use one or two types of resources to exchange for the other resource from high-utilization tasks that could bring more benefit to these cores.

(5) Phase 5 (Lines 16–18) checks the schedulability of the low-utilization tasks. If the low-utilization cores still cannot schedulable all the low-utilization tasks after the exchange, it is possible that they contain some tasks with relatively high utilization. Thus, the next iteration considers moving one more task to the set of high-utilization tasks. Otherwise, the task set is schedulable.

Algorithm 1: Holistic Resource Allocator

Input: τ: task set; N_{th}: total number of cores; N_{cp}: total number of cache partitions; N_{mp}: total number of MBW partitions; *maxKM*: the maximum iterations for KMeans; *maxPerm*: the maximum iterations.

Output: Schedulable or Unschedulable.

1 $\{\tau^{high}, \tau^{low}\} \leftarrow initHighTask(\tau, N_{cp}, N_{mp})$

2 **for** $m_{addH} = 0$ **up to** $size(\tau) - size(\tau^{high})$ **by** $+1$ **do**

3 $\quad \{\tau^{high}, \tau^{low}\} \leftarrow selectAddHT(\tau^{high}, \tau^{low}, m_{addH})$

4 $\quad \{\tau^{high}\} \leftarrow allocHR(\tau^{high}, N_{th}, N_{cp}, N_{mp})$

5 $\quad sched = checkSched(\tau^{high})$

6 $\quad$ **if** $sched = unschedulable$ **then break**;

7 $\quad \{N_{th}^{low}, N_{cp}^{low}, N_{mp}^{low}\} \leftarrow initLR(\tau^{high}, N_{th}, N_{cp}, N_{mp})$

8 $\quad$ sort($clusters \leftarrow clusterTasks(\tau^{low}, N_{th}, maxKM)$)

9 $\quad sched = unschedulable$

10 $\quad$ **for** $j = maxPerm$ **down to** 0 **by** -1 **do**

11 $\quad\quad perm\_clusters \leftarrow permute(clusters)$

12 $\quad\quad \{cores^{Sched}, cores^{USched}, sched\} \leftarrow$
$\quad\quad\quad CaMAllocLR(perm\_clusters, N_{th}^{low}, N_{cp}^{low}, N_{mp}^{low})$

13 $\quad\quad$ **if** $sched = schedulable$ **then break**;

14 $\quad\quad \tau^{high}, cores^{USched} \leftarrow resEx(\tau^{high}, cores^{USched})$

15 $\quad\quad cores^{low} = cores^{Sched} \cup cores^{USched}$

16 $\quad\quad sched = checkSched(cores^{low})$

17 $\quad\quad$ **if** $sched = schedulable$ **then break**;

18 $\quad$ **if** $sched = schedulable$ **then break**;

5.5.2 Procedures of the Algorithm

We now give details to the two main procedures in Algorithm 2.

Procedure *allocHR()* allocates resources to each high-utilization task for meeting its deadline. The minimum numbers of cores th_{min}, cache partitions cp_{min}, and MBW partitions mp_{min} for task τ_k are calculated such that $e_k(th_{min}, N_{cp}, N_{mp}) > d_k, e_k(N_{th}, cp_{min}, N_{mp}) > d_k$, and $e_k(N_{th}, N_{cp}, mp_{min}) > d_k$. After initializing the minimum resources, the procedure calculates the maximum benefit of allocating one instance of resource to one unschedulable task. For example, the benefit of allocation an additional core to τ_k with assignment (th_i, cp_i, mp_i) is calculated as $u_k(th_i + 1, cp_i, mp_i) - u_k(th_i, cp_i, mp_i)$. The benefit of allocating one cache or MBW partition can be calculated similarly. Among all the possible allocations with different available resources and different unschedulable tasks, the procedure will choose the one that results in the largest benefit (i.e., the largest reduction in utilization). Hence, this choice of allocation best utilizes the resources for high-utilization tasks.

Procedure *resEx()* tries to exchange resources between unschedulable cores partitioned with low-utilization tasks and schedulable high-utilization tasks. Note that although Procedure *allocHR()* makes good allocation decisions for high-utilization tasks, such a decision may not be globally optimal for low-utilization tasks. For example, the cache may only result in a slightly better benefit than MBW for high-utilization tasks, but *allocHR()* will assign all the available cache to high-utilization tasks. With no cache left, the low-utilization tasks cannot be scheduled on the remaining cores. At a high level, Procedure *resEx()* enumerates all valid resource exchange plans between a low-utilization core and a high-utilization task. An exchange will only happen if the high-utilization task remains schedulable and the low-utilization core has utilization benefits after the exchange.

Specifically, there are three steps in this procedure. First, if there are some empty cores not used by any low-utilization tasks, then each of the unschedulable low-utilization cores tries to use these empty cores to exchange for the cache and memory bandwidth

resources from each of the high-utilization tasks. This exchange is successful and will take place if both of them are schedulable after the exchange. Note that some cores can be empty while some other low-utilization cores are unschedulable. This is mainly because there does not remain at least one cache partition and one memory bandwidth partition to be associated with the empty core. Hence, when this happens, the empty cores can be used by high-utilization tasks that are already allocated with some cache and memory bandwidth. With extra cores, these high-utilization tasks may return some cache and memory bandwidth partitions, while being schedulable. These resources can then be used by the unschedulable low-utilization cores to improve their schedulability.

Second, if there is no empty core available, unschedulable low-utilization cores can try to use some of their cache and memory bandwidth resources to exchange one or more cores from high-utilization tasks. Lastly, we can also exchange the cache and memory bandwidth resources between low-utilization and high-utilization tasks. This may help balance these two types of resources and obtain the minimum number of cache partitions and memory bandwidth partitions needed for scheduling the low-utilization tasks on their cores. As a result, each core can be assigned the minimum resources.

5.5.3 Complexity of the Algorithm

We first discuss the complexity of the subprocedures and then summarize the total complexity of the algorithm. The following procedures iterate over all tasks and take $\mathcal{O}(m)$ time: *initHighTask()*, *selectAddHT()*, *checkSched()*, and *initLR()*. The *allocHR()* procedure enumerates all unschedulable high-utilization tasks for finding the one with the maximum benefit, and the number of times that this process repeats is at most the number of remaining resources, so this procedure takes $\mathcal{O}(m \cdot (N_{th} + N_{cp} + N_{mp}))$ time. The *sort()* procedure takes $\mathcal{O}(m \log m)$ time. Since the *maxKM* and *maxPerm* are predetermined constants, the *clusterTask()* procedure iterates over all clusters for all tasks for a constant number of iterations (i.e., $\mathcal{O}(m \cdot N_{th})$). The *permute()* procedure takes a constant time.

The complexity of the *CaMAllocLR()* procedure depends on the number of low-utilization tasks and the remaining resources. In the worst-case, all tasks are low-utilization, which takes $\mathcal{O}(\max\{m \log m, m \cdot N_{th}, N_{th} \cdot N_{cp}^2 \cdot N_{mp}^2\})$ time. The *resEx()* procedure takes $\mathcal{O}(m \cdot \max\{N_{th}, N_{cp}, N_{mp}\}^2)$ in the worst-case. Therefore, the entire algorithm has a complexity of $\mathcal{O}(m \cdot \max\{m \cdot (N_{th} + N_{cp} + N_{mp}), m^2 \log m \cdot N_{th}, N_{th} \cdot N_{cp}^2 \cdot N_{mp}^2, m \cdot \max\{N_{th}, N_{cp}, N_{mp}\}^2\})$.

Algorithm 2: GPUSched1(α, Q)

```
1 while Q is not empty do
2     if |a| ≥ αm then
3         τ_i := Dequeue(Q);
4         assign min(|a|, χ_i) processors to τ_i
5         offload τ_i to its assigned processors;
6         execute τ_i on its assigned processors;
```

5.6 Numerical Evaluation

In this section, we conducted numerical experiments to evaluate our proposed holistic resource allocation algorithms on task sets randomly generated using the profiling results of realistic parallel benchmark programs in Section 5.2.

5.6.1 Experimental Setup

Workload generation. To evaluate the scalability of our proposed algorithms, we consider four types of hardware systems. The smallest one has 13 cores with 10 cache partitions and 20 MBW partitions, which is consistent with our real hardware platform described in Section 5.2. A slightly larger one has twice the amount of resources, i.e., 26 cores with 20 cache partitions and 40 MBW partitions. We also consider the one with three times the resources (i.e., 39 cores with 30 cache partitions and 60 MBW partitions) and the largest

one with four times the resources (i.e., 52 cores with 40 cache partitions and 80 MBW partitions).

We vary the total reference utilization of task sets from 2 to the number of available cores m. For each total utilization, we randomly generate 200 task sets with the desired utilization. We generate a task's reference utilization uniformly at random from the utilization range from 0.2 to $0.5\sqrt{m}$, where m is the number of available cores. For each task, its WCET profile is randomly chosen from the empirical measurements of the 12 real-world benchmark programs described in Section 5.2. Each benchmark has an equal probability of being chosen unless otherwise specified. Due to the long profiling time discussed in Section 5.2.3, we were only able to measure each benchmark's execution times upon each resource allocation combination for one time, instead of multiple times for taking the worst-case value. Thus, the measured execution times have some variances. Hence, we refine the WCET profiles by making sure that the WCET upon a specific resource allocation is the same or larger than all the WCETs upon resource allocations that have strictly more allocated resources. Additionally, the measurements are performed on our real hardware platform with a limited number of resources (i.e., 13 cores with 10 cache partitions and 20 MBW partitions). In order to conduct larger-scale numerical experiments considering larger machines with more resources and more tasks, we set the WCETs upon more resources the same as the one upon 13 cores with 10 cache partitions and 20 MBW partitions.

The period (and implicit deadline) of a task is obtained by using its reference WCET to divide its reference utilization. For most of the experiment settings except for one, before adding a task to the task set, we also check whether this task can meet its implicit deadline when all the available resources of the hardware platform have been allocated to this task. If it is still not schedulable, then we do not add this task to the task set; otherwise, the task set will also become unschedulable no matter how resources are allocated to tasks.

5.6.2 Schedulability Performance

To the best of our knowledge, there are no existing solutions that address the core, cache, and memory bandwidth resource allocation problem for parallel real-time tasks. Therefore, in the first experiment, we evaluate our proposed holistic algorithm by comparing it against the optimal solution that is based on solving the mixed-integer nonlinear programming problem.

In this experiment, we consider four versions of the optimal. The first is the original MINLP formulation (denoted as OPT) that is directly solved by the solver in the SCIP Optimization Suite [36]. The next is the improved implementation (denoted as OPT-Imp). As discussed in Section 5.4, the implementation of the optimal is extremely inefficient. Even with our improved optimal variation, the running time is still very long. Moreover, when there is no valid allocation, the solver used by all the optimal solutions will not return any result. Therefore, we need to set a timeout for the optimal solutions. For each task set, it is deemed unschedulable if the optimal algorithm does not return any result by 15s, 1min, 10min, and 20min. Figure 5.6(a) shows the comparison results. The original implementation with 20min running time performs the worst, simply because directly solving the general MINLP is extremely hard. To verify that our implementation is correct, for task sets that are schedulable using our holistic algorithm, we initialize the corresponding variables in the MINLP formulation as the same as our generated allocation decision. Once doing that, the MINLP solver can return in a shorter time and verify that the encoded allocation decision passes all the optimization constraints. Compared with the improved implementation, we can see that our algorithm has comparable performance with the improved optimal algorithm with 1min running time limit and outperforms those that have shorter running time limits.

5.6.3 Running Time Efficiency

Although our holistic algorithm achieves slightly lower schedulability than the best performing optimal variation OPT (10min0.1s), it is significantly more efficient than any of the optimal variations. As shown in Figures 5.6(b) and 5.6(c), the average running times of all the optimal variations grow at an exponential rate. For example, the OPT variation that solves a single MINLP problem needs 4min on average to obtain solutions for task sets with reference utilizations that are as low as 3. For many schedulable task sets with reference utilizations that are larger than 4.5, OPT cannot return the valid resource allocation even if running for 2 hours.

The improved optimal variations have much faster running times, but they are still many orders of magnitude slower than the holistic algorithm. For instance, when the task set reference utilization is at least 5.5, OPT (10min0.1s) needs more than 1min on average to find the schedulable resource allocations. Its running time rapidly increases to 10min when the total reference utilization is 9.5. The fastest optimal variation OPT (10s0.1s) that has worse schedulability than the holistic algorithm takes about 1.5min to solve for task sets with reference utilization 10. In contrast, the average running times of our holistic algorithm are all below 24s. This computation efficiency gap between the optimal variations and the holistic algorithm increases when there are more available resources.

5.6.4 Impact of Different Benchmarks

As discussed in workload generation, in all of the other experiments, we only add a randomly generated benchmark task into the task set if this task can at least meet its deadline when monopolizing all the resources. Figure 5.7 motivates why it is necessary to do so. In particular, `BFS`, `Nbody`, `facesim`, and `MSF` are memory-intensive benchmarks. Because the reference utilization is defined as the WCET of a task on a single core with all the available cache and MBW partitions, these memory-sensitive tasks already have high speedup given all the MBW partitions. When their randomly generated utilizations are

high, allocating all the cores to them may not achieve sufficient additional speedup to allow them to meet their deadlines. In Figure 5.7, we do not check the schedulability of a task upon task set generation. Instead, we selectively exclude some of these 4 benchmarks in the task set generation. The differences in schedulability, thus, reflect the impact of these benchmarks.

To study the effect of different types of benchmarks, we also conduct an experiment where we generate task sets with only one type of benchmark and compare the schedulability results to the task sets with mixed types. Figure 5.8 shows that the fraction of schedulable task sets drastically drops to 0 for task sets with cache- and MBW-sensitive tasks. In contrast, when there are more cache- and MBW-insensitive tasks, there is a fraction of task sets schedulable at high total reference utilizations. The noticeable difference here also implicitly and partially verifies that our classification of benchmark programs indeed distinguishes the different characteristics of tasks.

5.6.5 Ablation Study of Our Algorithm

The impact of the different phases of our holistic algorithm can be observed in Figure 5.9(b). Specifically, version NoResEx disables the procedure *resEx()*, so there is not resource exchange between low- and high-utilization tasks. Version NoL2H disable procedure `selectAddHT()` and use a fixed threshold for classifying low- and high-utilization tasks. In contrast, NoMaxBene does not use the maximum benefit strategy in procedure `allocHR()` for finding the best resource to allocate to a high-utilization task and uses round-robin instead. We can see that procedure `allocHR()` and `selectAddHT()` both have large impacts on the schedulability of task sets, while procedure `resEx()` almost no impact for the task sets that select benchmarks uniformly at random in Figure 5.9(a). Based on our classification for benchmark programs, there are only 2 cache- and MBW-sensitive benchmarks out of the 12 benchmarks. When a task set has few or no such benchmarks, resource exchanges are unlikely to happen. To verify our hypothesis, we construct task

sets that select a cache- and MBW-sensitive benchmark with a probability twice the probability of selecting the other ones. As shown in Figure 5.9(b), with more cache- and MBW-sensitive benchmarks, resource exchange plays a more important role in finding a good resource allocation.

5.6.6 Impact of Platform Configurations and Task Parameters

Finally, we also vary the task parameters and platform configurations to evaluate their impact on the schedulability of task sets. As expected, increasing the number of available resources allows task sets with higher total reference utilizations to be schedulable. And the increase in schedulability is roughly proportional to the increase in the total resources, as revealed in Figure 5.10(a). On the other hand, different types of resources have different levels of impact on schedulability. As shown in Figure 5.10(b), decreasing the number of cache partitions by half has a smaller impact, compared to decreasing the number of memory bandwidth partitions by half. However, this is related to the characteristics of the randomly generated task sets. For example, if all tasks are insensitive to cache and memory bandwidth, the decrease in both resources will have little impact on schedulability. In Figure 5.10(c), we vary the range of tasks' reference utilizations. We can see that this change has a relatively small impact on the schedulability of task sets.

5.7 Empirical Evaluation

To demonstrate the practicality of our proposed framework on real hardware platforms and evaluate its empirical performance, we extend the federated scheduling system [83] to support cache and memory bandwidth partitioning. The modification to the federated scheduling is similar to the modification to benchmark programs as described in Section 5.2.

We first measure the system overhead of different resource allocation operations, including the core allocation operation via `pthread_setaffinity_np()`, the cache partition allocation operation via CAT, and the memory bandwidth allocation via Memguard.

Results presented in Table 5.3 show that the overhead of allocation cache partitions is significantly higher than partitioning cores and setting memory bandwidth reservations. Fortunately, our proposed framework follows the federated scheduling paradigm where the resource partitionings are performed only once before the execution of the task sets. Therefore, the high overhead only occurs once even before the execution of all tasks, does not impact the schedulability and efficiency of the task sets, and needs not to be incorporated into the analysis.

Table 5.3 Overhead Measurement

Operation	Average overhead (ms)	Maximum overhead (ms)
Core allocation	0.142	0.152
Cache partition allocation	569.1	570.8
Memory bandwidth allocation	4.4	4.5

We choose 3 representative benchmark programs (`RayCast`, `Swaptions`, and `RemDup`), one in each type, and measure their execution times of 50 runs on all 2,600 combinations of resource allocations. Since the measured maximum execution times of 50 runs may not be the true and safe WCET, we inflate the maximum value by 1.1 and use the inflated WCET for task set generation. We follow a similar procedure to that in Section 5.6 to randomly generated 100 task sets for each reference utilization on 13 cores. We run each task set for a duration that is equal to 50 times the longest period of the tasks in the set.

Although when generating task sets, we inflate the measured maximum execution time by 1.1. In the experiments, we would like to evaluate how the WCET estimates used by the holistic algorithm affect the schedulability of task sets. Hence, we use three types of WCET estimates: (1) the measured maximum execution time inflated by 1.1 — namely WCETx1.1; (2) the measured maximum execution time without any inflation — namely WCETx1.0; and (3) the regression results using 250 sampled maximum execution times inflated by 1.1 — WCETx1.1+fitting.

Note that both the greedy algorithm and the MINLP algorithms cannot generate a resource allocation decision for a task set, if the task set is deemed unschedulable by the respective algorithm. Additionally, it is not clear how to modify the optimal MINLP formulations to output any allocation decision for unschedulable task sets, as the optimization constraints are violated for these task sets. Therefore, one can only run the theoretically schedulable task sets to see if there is any deadline miss observed during the actual execution on the hardware platform. These results are presented in Figure 5.11 with legends WCETx1.0, WCETx1.1, and WCETx1.1+fitting. *In contrast to the numerical experiments in the previous section, here a task set is considered schedulable only if no deadline miss is observed during the actual execution of this task set on the hardware platform.*

We would also like to see how pessimistic the measured WCETs are when the workload is consolidated onto the platform following our proposed heuristic. To achieve this, we modified our holistic resource allocation strategy to generate one "reasonable" allocation decision, even when the task set is deemed unschedulable. In this way, we can execute the theoretically unschedulable task sets on the real platform to reveal the pessimism of the measured WCETs. Specifically, for every potential allocation decision tested during the process of the heuristic-based algorithm, we calculate the maximum of each core's utilization given this allocation decision and the task set. We record and output the decision that has the lowest maximum core utilization. Intuitively, this metric helps output the allocation decision that balances the load of each core. The observed schedulability of this best-effort approach (on top of the allocation decisions for the theoretically schedulable task sets) with WCETx1.0 estimates is denoted as WCETx1.0+best-effort in Figure 5.11.

We can observe that inflating the maximum execution times introduces a small amount of pessimism. A small number of task sets become unschedulable, because increasing the WCET estimate causes the need for more resources to meet deadlines.

Moreover, comparing the results between WCETx1.0 and WCETx1.0+best-effort, we can see that a few theoretically unschedulable task sets do not encounter any deadline miss during the actual execution.

Among the experimented task sets, we do not observe any deadline miss for both WCETx1.1 and WCETx1.0. In contrast, if we directly executing the resource allocation decisions made by the algorithm using WCETx1.1+fitting estimates, there will be two tasks missing their deadlines. This is because the estimation made by the regression algorithm over-optimistically predicts the WCETs for some resource allocations. When the holistic algorithm happens to choose these allocations, the involved tasks will miss deadlines. However, as discussed in Section 5.2, when regression with a small number of sampled data points is used for estimating the WCETs, profiling for the chosen resource allocation must be performed. If the profiling indicates that a task cannot meet its deadline given this resource allocation, a local refinement can be made by adding more resources to this task until the profiling shows that it can meet its deadline. For the particular two task sets in our experiments, we adopt this procedure and locally refine the allocation. The two task sets become schedulable. After this procedure, we do not observe any deadline miss for WCETx1.1+fitting.

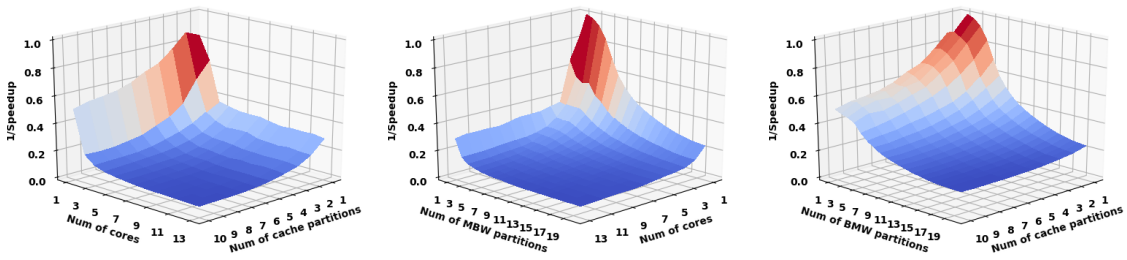
Note that the regression results can also be pessimistic. For example, there are a few task sets under reference utilization 3 that are deemed unschedulable using the WCETx1.1+fitting estimates. In principle, a local refinement can also be applied here to greedily search for potential resource allocation and validate it using profiling. Overall, the empirical experiments verify the efficiency and effectiveness of our proposed framework for parallel real-time applications.

5.8 Discussion and Future Work

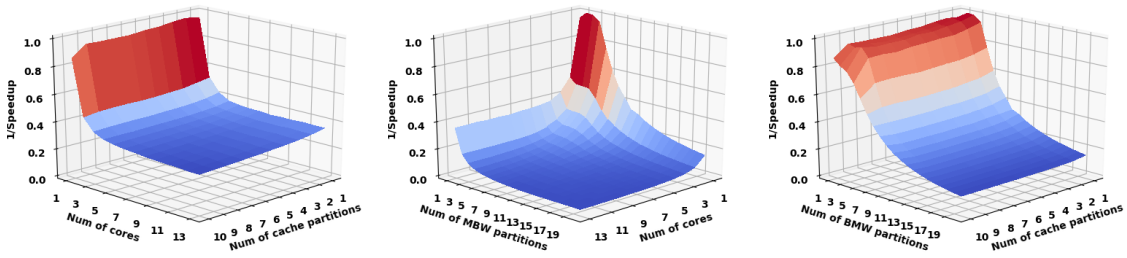
In future work, we plan to extend our insights and mechanisms to global scheduling algorithms and constrained deadline tasks. In addition to partitioning and isolating memory

bandwidth resources, incorporating the memory partitioning techniques into our framework is reserved for future work. We would also like to explore whether dynamically allocating cache and memory bandwidth resources can further improve the performance of parallel real-time systems.

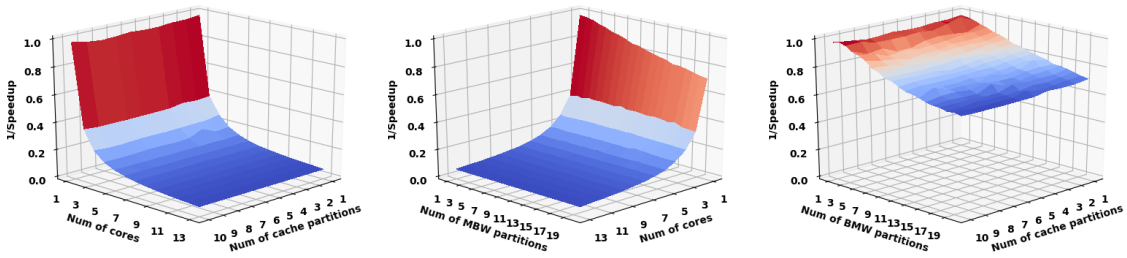
Furthermore, we will consider applying risk measures in parallel real-time scheduling, and Chapter 5 is just our first step to better achieve this. To briefly explain the idea, we will first assume a random inter-arrive time between two neighboring jobs of one task, which is assumed independent of all other jobs from the same or different tasks. So we will have a random response time, which is the time between a job is realsed and finished. Then the probability is introduced to the model so that we can apply risk measures, and our goal is to meet the deadlines of every task with very high probability (e.g., $\geq 99\%$ jobs finish before 10 ms). For example, we can compute the $p-$quantile of the response time to learn more information about how "far" is it from the deadline if one scheduler does not meet constraints. If multiple schedulers meet the constraints, to compare them, we can give more constraints by using multiple deadlines with different probabilistic constraints (e.g., $\geq 95\%$ jobs finish before 10 ms, $\geq 99\%$ jobs finish before 20 ms). We can also compare conditional tail expectation (CTE) of the response time, where CTE is the expectation of response time given response time is greater than the deadline. To make this easier, we can start with only considering the sequential tasks, and then takes the parallel tasks into account.



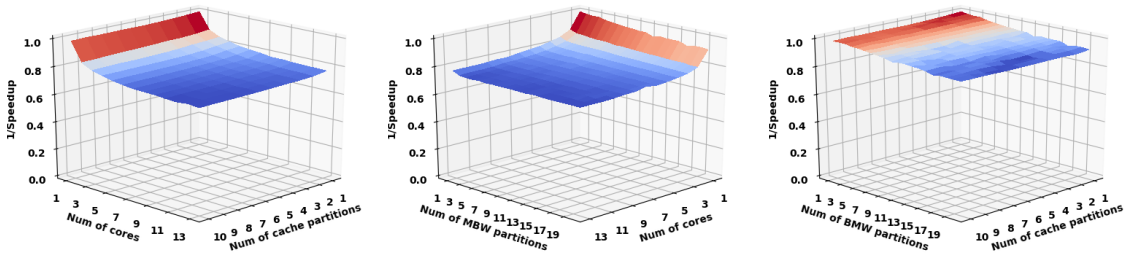
(a) RayCast: #bandwidth = 1. (b) RayCast: #cache = 1. (c) RayCast: #core = 1.



(d) BFS: #bandwidth = 1. (e) BFS: #cache = 1. (f) BFS: #core = 1.

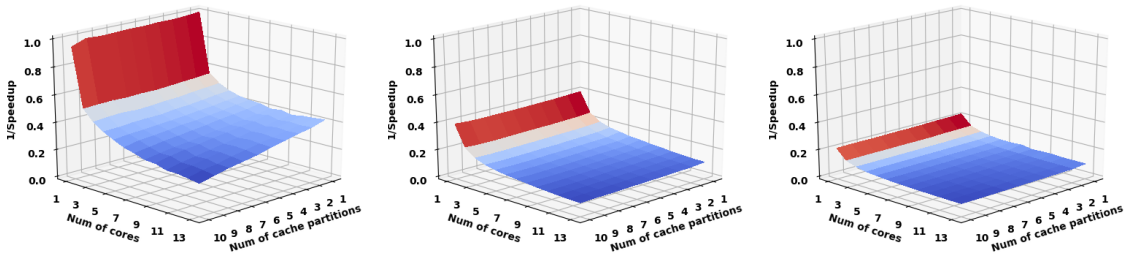


(g) Blackscholes: #bandwidth = 1. (h) Blackscholes: #cache = 1. (i) Blackscholes: #core = 1.



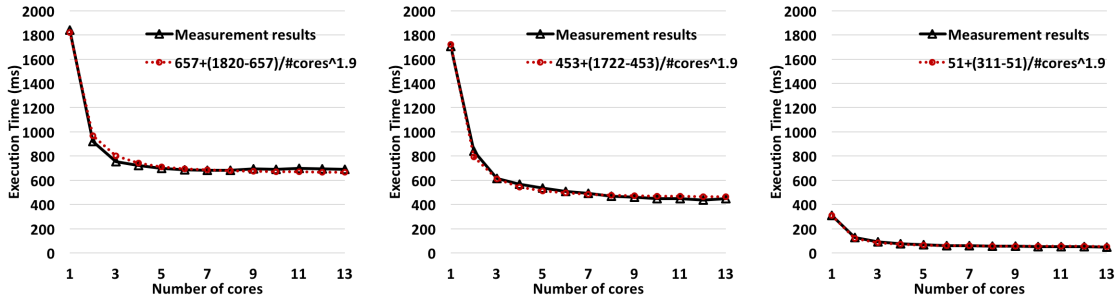
(j) Nbody: #bandwidth = 1. (k) Nbody: #cache = 1. (l) Nbody: #core = 1.

Figure 5.1 The slowdown (i.e., 1/speedup) of benchmark programs when assigning different numbers of cores, cache partitions, and memory bandwidth (MBW) partitions.

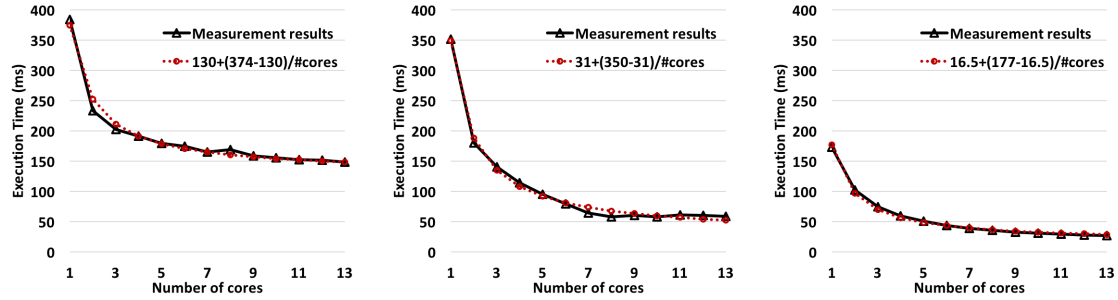


(a) Facesim: #bandwidth = 1. (b) Facesim: #bandwidth = 10. (c) Facesim: #bandwidth = 20.

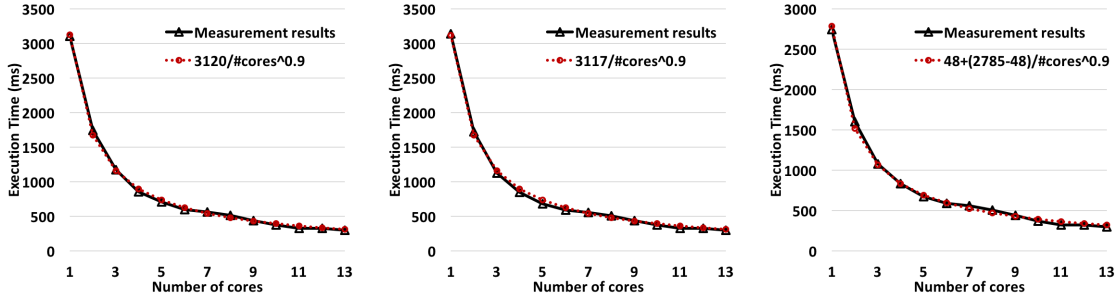
Figure 5.2 The slowdown (i.e., 1/speedup) of Facesim when assigning different numbers of cores, cache partitions, and memory bandwidth (MBW) partitions.



(a) RemDup: #cache=1, #MBW=1. (b) RemDup: #cache=10, #MBW=1. (c) RemDup: #cache=10, #MBW=20.

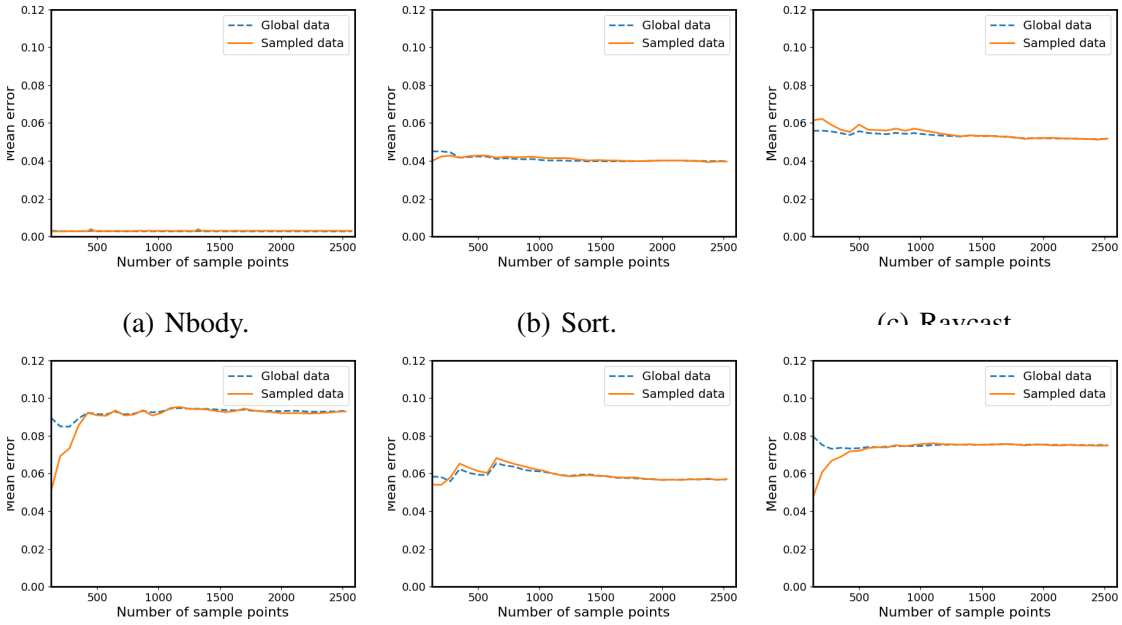


(d) Sort: #cache=1, #MBW=1. (e) Sort: #cache=10, #MBW=1. (f) Sort: #cache=10, #MBW=20.



(g) Fluid: #cache=1, #MBW=1. (h) Fluid: #cache=10, #MBW=1. (i) Fluid: #cache=10, #MBW=20.

Figure 5.3 Fitting the measured execution times of RemDup, Sort, and Fluid using nonlinear regression.



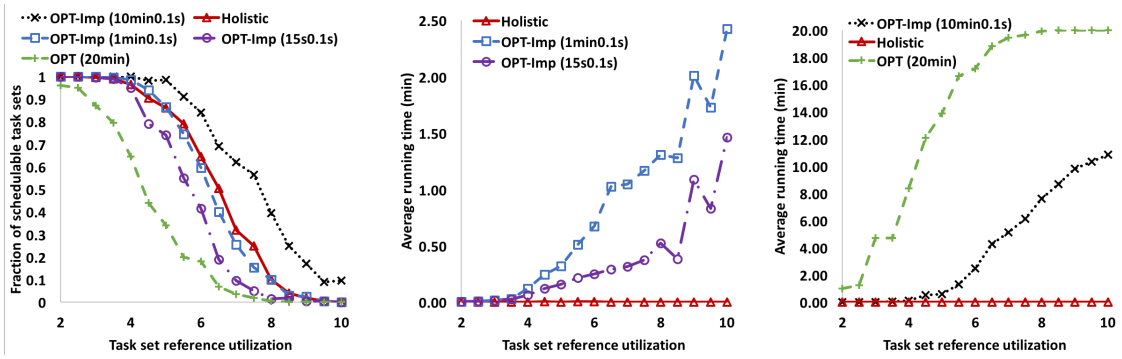
(a) Nbody. (b) Sort. (c) Raycast

(d) Dictionary. (e) Bodytrack. (f) Swaptions.

Figure 5.4 Mean relative error of fitting the measured execution times of different benchmarks when increasing the number of sampled data points used for the nonlinear regression: the solid line is the "global" MRE calculated using all the 2,600 data points, while the dashed line is the "local" MSE calculated using only the sampled data points.

	$\zeta_{i,j}$	Task 1 $\beta_1 = 0$	Task 2 $\beta_2 = 1$	Task 3 $\beta_3 = 0$	Task 4 $\beta_4 = 0$	$CPcore_j$	$MPcore_j$	Core Util
Core 1	$\gamma_1 = 0$				1	1	2	0.6
Core 2	$\gamma_2 = 1$		1					
Core 3	$\gamma_3 = 0$	1		1		4	8	0.8
Core 4	$\gamma_4 = 1$		1					
$CPtask_i$			1			N_{cp}=6		
$MPtask_i$			2				N_{mp}=12	
Task Util			0.7*2					

Figure 5.5 An example task set with 4 tasks on 4 cores with 6 cache partitions and 12 memory bandwidth partitions. The empty cells have a value of 0 for the corresponding variable.



(a) Schedulability comparison. (b) Average computation time. (c) Average computation time.

Figure 5.6 Comparison with the optimal variations for task sets on 13 cores with 10 cache and 20 MBW partitions.

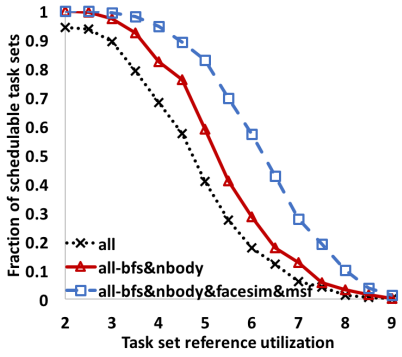


Figure 5.7 Experiments with a subset of benchmarks on 13 cores.

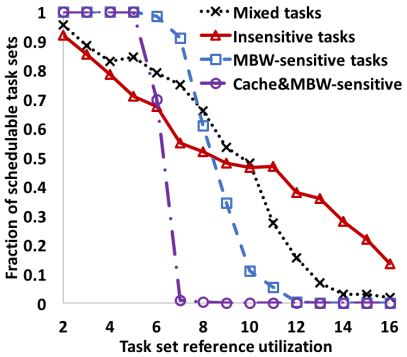
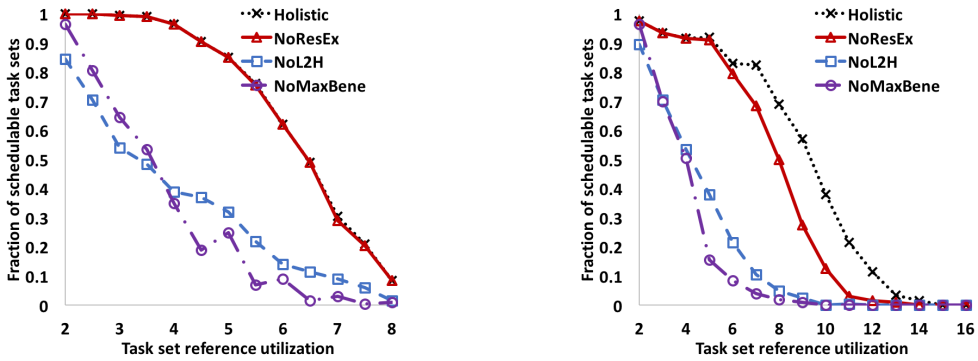


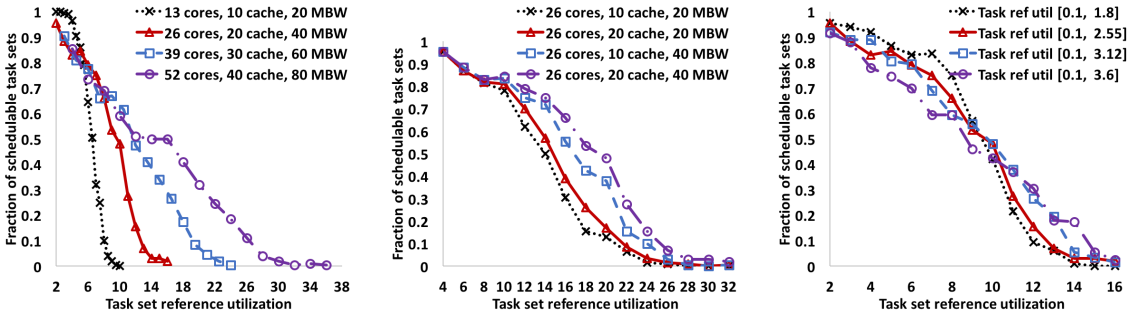
Figure 5.8 Experiments with specific type of benchmarks on 26 cores.



(a) Fewer cache- and MBW-sensitive tasks.

(b) More cache- and MBW-sensitive tasks.

Figure 5.9 Ablation study of holistic algorithm on task sets with different percentages of cache- and MBW-sensitive tasks on 26 cores, 20 cache partitions, and 40 MBW partitions.



(a) Increase the number of all resources.

(b) Change the number of some resources.

(c) Change the task utilization range.

Figure 5.10 Fraction of schedulable task sets with different task set generation parameters.

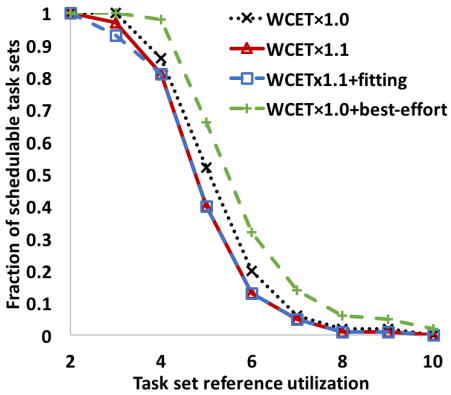


Figure 5.11 Empirical experiments on 13 cores.

CHAPTER 6

CONCLUSION

This research work mainly discusses about the computation of risk measures in finance and parallel real-time scheduling.

In Chapters 2 and 3, we used multiple methods to estimate one risk measure: quantile. First, we compared two approaches for quantile estimation via RQMC in an asymptotic setting where the number of randomizations for RQMC grows large but the size of the low-discrepancy point set remains fixed. Unfortunately, the first estimator converges to the wrong value as $r \to \infty$, leading to poor coverage. In contrast, the second quantile estimator does converge in the desired value ξ. We are currently working on formulating asymptotic regimes in which both $m \to \infty$ and $r \to \infty$. We are also working on comparing different real time schedulers based on quantile. Then we have shown how combining CMC with RQMC can have a synergistic effect in improving the accuracy of quantile estimators in a setting where observations are obtained from simulation. The synergy comes from CMC replacing the empirical CDF (a step function) by a smooth, RQMC-friendly, CDF estimator. This can make the subsequent RQMC improvement more substantial than when RQMC is applied alone. As $f(\xi)$ appears in the denominator of the asymptotic variance in (3.21), constructing a confidence interval for ξ may also benefit from applying CMC+RQMC for density estimation. In follow-up work, we intend to provide sufficient conditions under which we can prove that the MSE of the quantile estimator converges at a faster rate than the canonical $O(1/n)$. We also want to experiment with a larger variety of examples.

In Chapter 4, we analyzed different approaches to estimate another risk measure: economic captical. The economic capital is is used to determine capital levels (e.g., [24]). Defined as the difference between the p-quantile ξ and the mean μ of the loss distribution, the EC in practice takes $p \approx 1$, in which case SRS is ineffective in estimating ξ. But

applying IS to estimate both ξ and μ can be detrimental, and MSIS instead estimates only ξ with IS, and independently estimates μ via SRS. We also consider ISDM and DE estimators of η. Our asymptotic theory (Theorem 8) and numerical results (Section 4.6) show that MSIS can outperform the other approaches. While our theoretical results for EC estimators for the i.i.d. sum model (Section 4.5) provide deep insights for problems in rare-event simulation and financial risk management, they also have implications for techniques that reuse simulation data ([86], [27]), which is also called "green simulation" ([31]). To estimate mean performances when parameters of underlying distributions in the same simulation model differ across experiments, green simulation reuses outputs from previous experiments by weighting them with likelihood ratios. In estimating a mean, as has been the focus of green simulation, our Theorem 6 shows that IS can result in estimators with an extremely large variance when a single simulation run requires generating many independent random variables, so resuing simulation data through likelihood ratios may be less effective in such contexts. [31] and [27] further apply (a slight variation of) ISDM in green simulations when estimating mean performances, and our Theorem 5 also reveals that ISDM can be quite effective to control the variance.

Chapter 5 presents a holistic resource allocation strategy for parallel real-time tasks executing on multicore systems that share cache and memory bandwidth resources. Our strategy integrates existing cache partitioning and memory bandwidth regulation mechanisms and leverages results from resource allocation for sequential tasks and federated scheduling for parallel tasks. Based on the insights obtained from empirical evaluations of real-world parallel benchmarks, we develop an approach for parallel real-time tasks to improve the practicality of measurement-based models. The numerical evaluation and proof of concept implementation demonstrate that our proposed framework is efficient and practical. In the end, the discussion about the application of risk measures for real-time scheduling is given for future work.

APPENDIX A

FURTHER NUMERICAL STUDY OF RELATIVE ERROR AND ITS APPROXIMATION

Recall that Section 4.6.1 presented numerical results for the RE and its approximation RĚ for the model in Section 4.5 when the i.i.d. summands have marginal distribution G_0 that is exponential. We now present some additional results for G_0 as normal $N(1,1)$ and Erlang ($s = 8$ stages, scale parameter 1).

For G_0 as $N(1,1)$ (resp., Erlang), Figure A.1 (resp., Figure A.2) plots the exact RE for estimators of η, ξ, and μ, and also the exact RE and its approximation RĚ for estimators of η. These two figures mostly exhibit the same basic trends that we saw in Figure 4.1 when G_0 is exponential: the η estimators have exponentially increasing RE as m grows for SRS, IS$(\theta_\star)$, and DE$(\theta_\star)$, decreasing RE for MSIS$(\theta_\star)$ and ISDM$(\theta_\star)$, and RE and RĚ for the same method generally match up well. But one difference is that for RE$[\eta]$ for large m, SRS is the worst for exponential G_0, whereas IS$(\theta_\star)$ is the worst for G_0 as $N(1,1)$ and Erlang. Also, as noted in the two paragraphs after Theorem 8, the behavior of RE$[\eta]$ for DE$(\theta_\star)$ is governed by the *worst* of the SRS and IS$(\theta_\star)$ estimators of ξ and μ, as seen in Figure 4.1 and Figures A.1 and A.2.

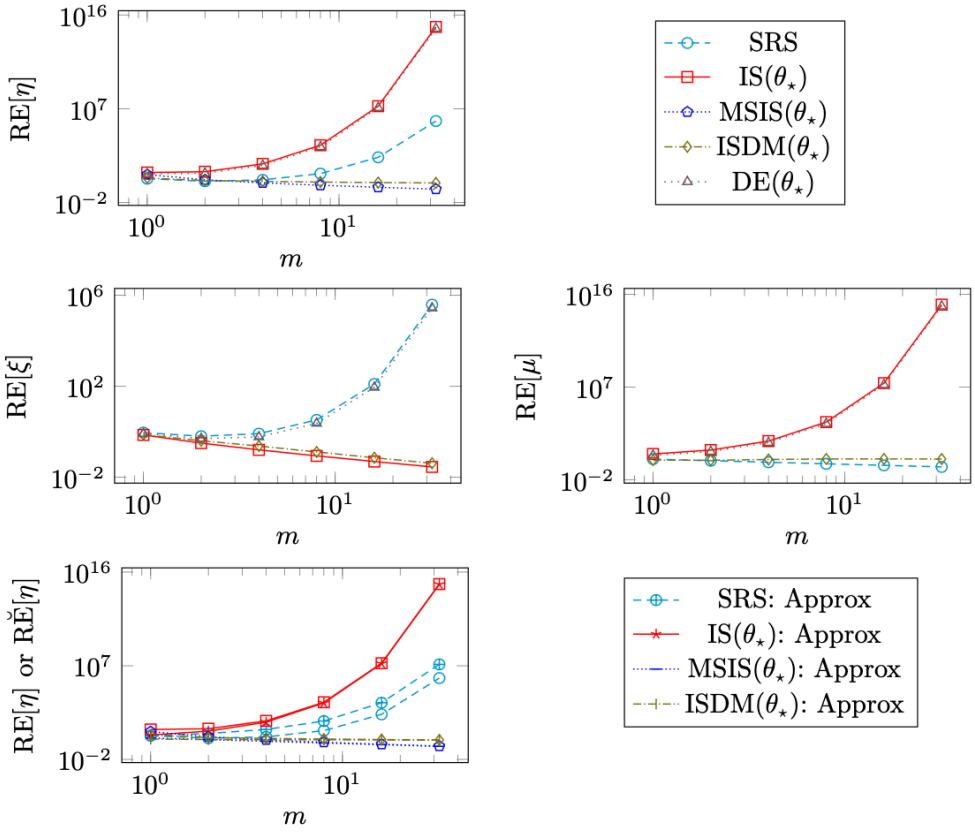


Figure A.1 For G_0 as $N(1,1)$ and $\beta = 1.1$ in (4.31), the log-log plots show the RE and its approximation RĚ, computed numerically (i.e., not estimated via simulation), as functions of the dimension m. In the top two rows, the plots display the exact RE of estimators of the EC η (top left panel), the p-quantile ξ (middle row, left panel), and the mean μ (middle row, right panel). The bottom left panel shows $\mathrm{RE}[\eta]$ and its approximation $\mathrm{R\check{E}}[\eta]$. The middle panels do not give results for $\mathrm{MSIS}(\theta_\star)$, which uses $\mathrm{IS}(\theta_\star)$ (resp., SRS) to estimate ξ (resp., μ).

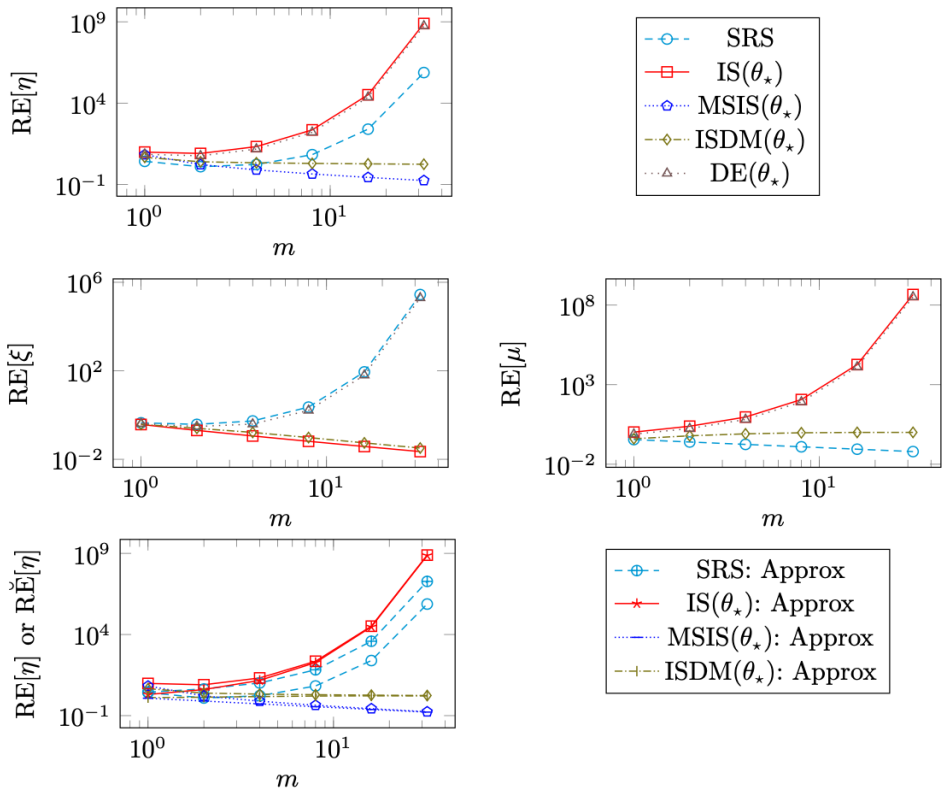


Figure A.2 For G_0 as Erlang ($s = 8$ stages, scale parameter 1) and $\beta = 1.1$ in (4.31), the log-log plots show RE and its approximation RĚ, computed numerically (i.e., not simulation), as functions of dimension m. In the top two rows, the plots display the exact RE of estimators of the EC η (top left panel), the p-quantile ξ (middle row, left panel), and the mean μ (middle row, right panel). The bottom left panel shows RE$[\eta]$ and its approximation RĚ$[\eta]$. The middle panels do not give results for MSIS$(\theta_\star)$, which uses IS$(\theta_\star)$ (resp., SRS) to estimate ξ (resp., μ).

APPENDIX B

PROOF OF THEOREM 2

Reference [37] proves Equations (4.6) and (4.7) hold when $f(\xi) > 0$. Further assuming that F is twice differentiable at ξ, [62] (see also p. 100 of [117]) derives the exact rate of convergence of R_n, given in (4.8), improving on the original result of [12] (also see Theorem 2.5.1 of [117]). In all cases, putting (4.6), (4.4), and (4.3) into (4.5) then leads to

$$\widehat{\eta}_{\mathrm{SRS},n} = \xi - \frac{\widehat{F}_{\mathrm{SRS},n}(\xi) - p}{f(\xi)} + R_n - \widehat{\mu}_{\mathrm{SRS},n} = \xi - \mu - \frac{1}{f(\xi)} \left[\frac{1}{n} \sum_{i=1}^{n} I(Y_i \leq \xi) - p \right] + R_n - \frac{1}{n} \sum_{i=1}^{n} Y_i + \mu,$$

which equals (4.9), establishing part (i).

We next prove part (ii). Rearranging (4.9) and scaling by $\sqrt{n}$ leads to

$$\sqrt{n} \left[\widehat{\eta}_{\mathrm{SRS},n} - \eta \right] = -\sqrt{n} \left[\frac{1}{n} \sum_{i=1}^{n} \left(\left[\frac{1}{f(\xi)} I(Y_i \leq \xi) + Y_i \right] - \left[\frac{p}{f(\xi)} + \mu \right] \right) \right] + \sqrt{n} R_n. \quad \text{(B.1)}$$

Let $A = [I(Y \leq \xi)/f(\xi)] + Y$ and $A_i = [I(Y_i \leq \xi)/f(\xi)] + Y_i$. Note that A_i, $i = 1, 2, \ldots, n$, are i.i.d. copies of A, where $\phi \equiv E[A] = [p/f(\xi)] + \mu$ because $f(\xi) > 0$ ensures $E[I(Y \leq \xi)] = F(\xi) = p$. Hence, the right side of (B.1) equals $-\sqrt{n}[\frac{1}{n}\sum_{i=1}^{n} A_i - \phi] + \sqrt{n} R_n$. Also, $\mathrm{Var}[I(Y \leq \xi)] = p(1-p) = \chi^2_{\mathrm{SRS}}$ implies

$$\mathrm{Var}[A] = \mathrm{Var}[I(Y \leq \xi)/f(\xi)] + \mathrm{Var}[Y] + 2\mathrm{Cov}[I(Y \leq \xi)/f(\xi), Y] = \zeta^2_{\mathrm{SRS}}$$

by (4.10) because $\gamma_{\mathrm{SRS}} = E[I(Y \leq \xi)Y] - p\mu$. We assumed that $\sigma^2_{\mathrm{SRS}} < \infty$ and $f(\xi) > 0$, so $\zeta^2_{\mathrm{SRS}} < \infty$ by the Cauchy-Schwarz inequality. Thus, the ordinary CLT ensures that

$$-\sqrt{n} \left[\frac{1}{n} \sum_{i=1}^{n} A_i - \phi \right] \Rightarrow N(0, \zeta^2_{\mathrm{SRS}}) \quad \text{as } n \to \infty. \quad \text{(B.2)}$$

As the limit in (4.7) is deterministic, the left sides of (B.2) and (4.7) jointly converge to their respective limits by Theorem 11.4.5 of [134]. Hence, applying the continuous-mapping theorem (e.g., Theorem 3.4.3 of [134]), we have that (4.10) holds. |

APPENDIX C

PROOF OF THEOREM 3

Put Equations (4.16) and (4.12) into (4.17) and use (4.16). This establishes part (i).

We next prove part (ii). The sum in (4.18) has i.i.d. summands, where each summand has mean 0 and variance ζ_{IS}^2. From the first part of each summand, we have that $\text{Var}_{\widetilde{G}}[1 - I(c(\mathbf{X}) > \xi)L(\mathbf{X})] = \text{Var}_{\widetilde{G}}[I(c(\mathbf{X}) > \xi)L(\mathbf{X})] = E_{\widetilde{G}}[(I(c(\mathbf{X}) > \xi)L(\mathbf{X}))^2] - (1-p)^2$, and

$$E_{\widetilde{G}}[I(c(\mathbf{X}) > \xi)L^2(\mathbf{X})] \leq E_{\widetilde{G}}[I(c(\mathbf{X}) > \xi - \lambda)L^{2+\varepsilon}(\mathbf{X})] < \infty \tag{C.1}$$

by (4.15), so $\chi_{\text{IS}}^2 < \infty$. Also, we assumed that $\sigma_{\text{IS}}^2 < \infty$, so the Cauchy-Schwarz inequality ensures that γ_{IS} is finite, implying the same is true for ζ_{IS}^2 because $f(\xi) > 0$. Thus, part (ii) holds by (4.18) and Slutsky's theorem (e.g., p. 19 of [117]). |

APPENDIX D

PROOF OF THEOREM 4

Put Equation (4.16) with $n_1 = \delta n$ replacing n into Equation (4.21) to get

$$\widehat{\eta}_{\mathrm{MSIS},n} = \xi - \mu - \frac{\widehat{F}_{\mathrm{IS},\delta n}(\xi) - p}{f(\xi)} + \tilde{R}_{n,\delta} - \widehat{\mu}_{\mathrm{SRS},(1-\delta)n} + \mu,$$

from which Equation (4.22) follows from Equation (4.16). This establishes part (i).

We next prove part (ii). Rearrange (4.22) and scale by $\sqrt{n}$ to get

$$\sqrt{n}\left[\widehat{\eta}_{\mathrm{MSIS},n} - \eta\right] = -\frac{\sqrt{n}}{f(\xi)}\left[\widehat{F}_{\mathrm{IS},\delta n}(\xi) - p\right] - \sqrt{n}\left[\widehat{\mu}_{\mathrm{SRS},(1-\delta)n} - \mu\right] + \sqrt{n}\tilde{R}_{n,\delta}.$$

As we showed in (C.1), (4.15) implies that $\chi^2_{\mathrm{IS}} < \infty$, so $f(\xi) > 0$ ensures that

$$\sqrt{n}[\widehat{F}_{\mathrm{IS},\delta n}(\xi) - p] \Rightarrow N_1' \sim N\left(0, \frac{\chi^2_{\mathrm{IS}}}{\delta f^2(\xi)}\right), \tag{D.1}$$

where the δ appears in the denominator of the asymptotic variance because the left side of (D.1) scales by $\sqrt{n}$ rather than $\sqrt{n_1}$, and the sample size used to construct $\widehat{F}_{\mathrm{IS},\delta n}$ is $n_1 = \delta n$. Then

$$\sqrt{n}[\widehat{\mu}_{\mathrm{SRS},(1-\delta)n} - \mu] \Rightarrow N_2' \sim N\left(0, \frac{\sigma^2_{\mathrm{SRS}}}{1-\delta}\right) \quad \text{as } n \to \infty \tag{D.2}$$

since $\sigma^2_{\mathrm{SRS}} < \infty$, where the $1-\delta$ appears in the denominator of the asymptotic variance because the scaling in (D.2) is $\sqrt{n}$ rather than $\sqrt{n_2}$. Under MSIS, $\widehat{\mu}_{\mathrm{SRS},(1-\delta)n}$ is independent of $\widehat{\xi}_{\mathrm{IS},\delta n}$ and $\widehat{F}_{\mathrm{IS},\delta n}$, guaranteeing the joint convergence of (D.1) and (D.2) as $n \to \infty$ by Theorem 11.4.4 of [134]. Moreover, because the limit in (4.22) is deterministic, it follows that

$$\left(\sqrt{n}\left[\widehat{F}_{\mathrm{IS},\delta n}(\xi) - p\right], \sqrt{n}\left[\widehat{\mu}_{\mathrm{SRS},(1-\delta)n} - \mu\right], \sqrt{n}\tilde{R}_{n,\delta}\right) \Rightarrow (N_1', N_2', 0) \quad \text{as } n \to \infty$$

by Theorem 11.4.5 of [134], where N_1' and N_2' are independent. Finally, applying the continuous-mapping theorem completes the proof. |

APPENDIX E

PROOF OF THEOREM 5

By Equation (4.25), we have $\widehat{\eta}_{\mathrm{DE},n} = \upsilon_1 \widehat{\xi}_{\mathrm{IS},\delta n} + \upsilon_1' \widehat{\xi}_{\mathrm{SRS},(1-\delta)n} - \upsilon_2 \widehat{\mu}_{\mathrm{IS},\delta n} - \upsilon_2' \widehat{\mu}_{\mathrm{SRS},(1-\delta)n}$, where $\widehat{\xi}_{\mathrm{IS},\delta n}$ and $\widehat{\mu}_{\mathrm{IS},\delta n}$ are from Equation (4.17), and $\widehat{\xi}_{\mathrm{SRS},(1-\delta)n}$ and $\widehat{\mu}_{\mathrm{SRS},(1-\delta)n}$ are from (4.5). Use the corresponding expressions for $\widehat{\xi}_{\mathrm{IS},\delta n}$ and $\widehat{\mu}_{\mathrm{IS},\delta n}$ in (4.18), and expressions for $\widehat{\xi}_{\mathrm{SRS},(1-\delta)n}$ and $\widehat{\mu}_{\mathrm{SRS},(1-\delta)n}$ in (4.9), and then rearrange these expressions to get (4.26).

To prove part (ii), we use (4.25) to split $\sqrt{n}[\widehat{\eta}_{\mathrm{DE},n} - \eta]$ into terms based on IS and terms based on SRS, which will be analyzed separately. For the IS estimators, modify the proof of Theorem 3(ii) to get the CLT

$$\sqrt{n}\left(\left[\upsilon_1 \widehat{\xi}_{\mathrm{IS},\delta n} - \upsilon_2 \widehat{\mu}_{\mathrm{IS},\delta n}\right] - [\upsilon_1 \xi - \upsilon_2 \mu]\right) \Rightarrow N_1 \tag{E.1}$$

as $n \to \infty$, where $N_1 \sim N(0, \psi^2_{\mathrm{IS},\delta,\upsilon_1,\upsilon_2})$ and

$$\psi^2_{\mathrm{IS},\delta,\upsilon_1,\upsilon_2} = \frac{1}{\delta}\left[\upsilon_1^2 \frac{\chi^2_{\mathrm{IS}}}{f^2(\xi)} + \upsilon_2^2 \sigma^2_{\mathrm{IS}} - 2\upsilon_1 \upsilon_2 \frac{\gamma_{\mathrm{IS}}}{f(\xi)}\right]. \tag{E.2}$$

Similarly, for the SRS estimators, we modify the proof of Theorem 2(ii) to get the CLT

$$\sqrt{n}\left(\left[\upsilon_1' \widehat{\xi}_{\mathrm{SRS},(1-\delta)n} - \upsilon_2' \widehat{\mu}_{\mathrm{SRS},(1-\delta)n}\right] - \left[\upsilon_1' \xi - \upsilon_2' \mu\right]\right) \Rightarrow N_2 \tag{E.3}$$

as $n \to \infty$, where $N_2 \sim N(0, \psi^2_{\mathrm{SRS},\delta,\upsilon_1,\upsilon_2})$ and

$$\psi^2_{\mathrm{SRS},\delta,\upsilon_1,\upsilon_2} = \frac{1}{1-\delta}\left[\upsilon_1'^2 \frac{\chi^2_{\mathrm{SRS}}}{f^2(\xi)} + \upsilon_2'^2 \sigma^2_{\mathrm{SRS}} + 2\upsilon_1' \upsilon_2' \frac{\gamma_{\mathrm{SRS}}}{f(\xi)}\right]. \tag{E.4}$$

Hence, $\sqrt{n}[\widehat{\eta}_{\mathrm{DE},n} - \eta]$ equals the sum of the left sides of (E.1) and (E.3) by (4.25). The estimators in (E.1) and (E.3) are independent, so the CLTs in (E.1) and (E.3) hold jointly with N_1 and N_2 independent [134, Theorem 11.4.5]. Thus, the continuous-mapping theorem [134, Theorem 3.4.3] implies $\sqrt{n}[\widehat{\eta}_{\mathrm{DE},n} - \eta] \Rightarrow N_1 + N_2 \sim N(0, \psi^2_{\mathrm{IS},\delta,\upsilon_1,\upsilon_2} + \psi^2_{\mathrm{SRS},\delta,\upsilon_1,\upsilon_2})$ as $n \to \infty$, proving (4.27).

Next we are going to prove for fixed $\delta \in (0,1)$, the optimal value of υ_1 and υ_2 is as in (4.28). The partial derivative of ζ^2_{DE} with respect to υ_1 is $\frac{\partial \zeta^2_{\mathrm{DE}}}{\partial \upsilon_1} = 2\mathrm{V}^{(\xi)}_{\mathrm{IS}} \upsilon_1 - 2\mathrm{V}^{(\xi)}_{\mathrm{SRS}}(1 -$

$\upsilon_1)+2(-C_{IS}\upsilon_2-C_{SRS}(1-\upsilon_2))$. The partial derivative of ζ^2_{DE} with respect to υ_2 is $\frac{\partial\zeta^2_{DE}}{\partial\upsilon_2}=2V^{(\mu)}_{IS}\upsilon_2-2V^{(\mu)}_{SRS}(1-\upsilon_2)+2(-C_{IS}\upsilon_1-C_{SRS}(1-\upsilon_1))$. By setting these two equations equal to 0 and solving, then we get $(\upsilon^*_1,\upsilon^*_2)$ in (4.28). |

APPENDIX F

PROOF OF THEOREM 6

Part (i): SRS When we apply SRS, (4.35) holds because $c(\mathbf{X}) = \sum_{j=1}^{m} X_j$, where $X_1, X_2, \ldots, X_m$ are i.i.d. with variance σ_0^2. As $\mu = m\mu_0$, we see that (4.36) follows, completing the proof of (i).

Part (ii): IS(θ) Now consider IS(θ) with $\widetilde{G}_\theta$ satisfying Assumption 2, so $\theta \in \Delta^\circ$, and we further assumed that $-\theta \in \Delta^\circ$. By (4.11) and a change of measure, we can write the variance in (4.37) as

$$\sigma^2_{\mathrm{IS}(\theta)} = E_{\widetilde{G}_\theta}[c^2(\mathbf{X})L^2_\theta(\mathbf{X})] - \mu^2 = E_G[c^2(\mathbf{X})L_\theta(\mathbf{X})] - (m\mu_0)^2, \tag{F.1}$$

giving the second term of (4.37) in both cases ($m = 1$ and $m \geq 2$). By (4.30), the likelihood ratio in (F.1) is

$$L_\theta(\mathbf{X}) = \prod_{j=1}^{m} \frac{\mathrm{d}G_0(X_j)}{\mathrm{d}\widetilde{G}_{0,\theta}(X_j)} = \exp\Big(mQ_0(\theta) - \theta \sum_{j=1}^{m} X_j\Big) = [M_0(\theta)]^m e^{-\theta c(\mathbf{X})} \tag{F.2}$$

because $Q_0(\theta) = \ln M_0(\theta)$, so the second-moment term in (F.1) becomes

$$E_G[c^2(\mathbf{X})L_\theta(\mathbf{X})] = [M_0(\theta)]^m E_G\left[c^2(\mathbf{X})e^{-\theta c(\mathbf{X})}\right]. \tag{F.3}$$

We next will show that (F.3) equals the first term in (4.37) in both cases ($m = 1$ and $m \geq 2$). We do this via derivatives of the MGFs and CGFs of $X_j \sim G_0$ and $c(\mathbf{X}) \sim F_m$. Let $M_{F_m}(\theta) = E_G[e^{\theta c(\mathbf{X})}]$, $\theta \in \mathbb{R}$, be the MGF of $Y = c(\mathbf{X}) = \sum_{j=1}^{m} X_j$. As the components of $\mathbf{X}$ are i.i.d., we have

$$M_{F_m}(\theta) = E_G\left[\prod_{j=1}^{m} e^{\theta X_j}\right] = \prod_{j=1}^{m} E_0\left[e^{\theta X_j}\right] = [M_0(\theta)]^m, \tag{F.4}$$

so $M_{F_m}(\theta) < \infty$ for $\theta \in \Delta^\circ$. We assumed that $\pm\theta \in \Delta^\circ$, in which case $M_0(\theta)$ and $M_0(-\theta)$ have derivatives of all orders [17, p. 278], and the same holds for $M_{F_m}(\theta)$ and $M_{F_m}(-\theta)$ by

(F.4). The second derivative of M_{F_m} satisfies $M''_{F_m}(\theta) = E_G\left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} e^{\theta c(\mathbf{X})}\right] = E_G[c^2(\mathbf{X}) e^{\theta c(\mathbf{X})}]$, so (F.3) becomes

$$E_G[c^2(\mathbf{X}) L_\theta(\mathbf{X})] = [M_0(\theta)]^m M''_{F_m}(-\theta). \tag{F.5}$$

For $m = 1$, (F.4) implies that $M''_{F_m}(\theta) = M''_0(\theta)$, and putting this into (F.5) establishes the first case of (4.37). For $m \geq 2$, we use (F.4) to express the first derivative $M'_{F_m}(\theta) = m[M_0(\theta)]^{m-1} M'_0(\theta)$ and

$$\begin{aligned} M''_{F_m}(\theta) &= m(m-1)[M_0(\theta)]^{m-2}[M'_0(\theta)]^2 + m[M_0(\theta)]^{m-1} M''_0(\theta) \\ &= m[M_0(\theta)]^m \left[(m-1)\left(\frac{M'_0(\theta)}{M_0(\theta)}\right)^2 + \frac{M''_0(\theta)}{M_0(\theta)}\right] \\ &= m[M_0(\theta)]^m \Big[m[Q'_0(\theta)]^2 + Q''_0(\theta)\Big] \end{aligned} \tag{F.6}$$

because the first two derivatives of the CGF $Q_0(\theta) = \ln M_0(\theta)$ are

$$Q'_0(\theta) = \frac{M'_0(\theta)}{M_0(\theta)} \quad \text{and} \quad Q''_0(\theta) = \frac{M''_0(\theta)}{M_0(\theta)} - \left[\frac{M'_0(\theta)}{M_0(\theta)}\right]^2. \tag{F.7}$$

As a consequence, substituting (F.6) in (F.5) yields

$$E_G[c^2(\mathbf{X}) L_\theta(\mathbf{X})] = m[M_0(\theta) M_0(-\theta)]^m \Big[m[Q'_0(-\theta)]^2 + Q''_0(-\theta)\Big] \tag{F.8}$$

for $m \geq 2$, giving the first term in the second case of (4.37). (When $\theta = 0$, which corresponds to SRS, we have $\widetilde{G}_\theta = G$ and $M_0(0) = \alpha(0) = 1$. Also, $M'_0(0) = \mu_0$ and $M''_0(0) = E_0[X_j^2]$, so (4.37) equals (4.35) by (F.7).)

To prove the inequalities in (4.38), we first show that

$$Q''_0(\theta) > 0, \quad \text{for each } \theta \in \Delta^\circ. \tag{F.9}$$

For $\theta \in \Delta^\circ$, the exponential twist $\widetilde{G}_{0,\theta}$ in (4.30) of G_0 has variance $Q''_0(\theta)$ (and mean $Q'_0(\theta)$) [29, pp. 72–73]. Because G_0 is nondegenerate (Assumption 1), the same holds for $\widetilde{G}_{0,\theta}$ as they share the same support by (4.30), so (F.9) is true. But $\pm\theta \in \Delta^\circ$, so (F.9) establishes the second inequality in (4.38).

We next show $\alpha(\theta) \equiv M_0(\theta)M_0(-\theta) \geq 1$ in (4.38) for all θ such that $\pm\theta \in \Delta^\circ$. When $\theta = 0$, we have that $M_0(0) = 1$, so the first relation in (4.38) holds in this case. Now assume that $\theta \neq 0$. Then (F.9) implies that $Q_0(\theta)$ is strictly convex on Δ°; also see, e.g., [29, p. 73]. Hence, Jensen's inequality yields

$$\alpha(\theta) = \exp\left[2\left(\frac{1}{2}Q_0(\theta) + \frac{1}{2}Q_0(-\theta)\right)\right] > \exp\left[2Q_0\left(\frac{1}{2}\theta - \frac{1}{2}\theta\right)\right] = e^{2Q_0(0)} = 1 \tag{F.10}$$

because $Q_0(0) = 0$, proving the first inequality of (4.38).

We next verify (4.39) for any $\theta \neq 0$ with $\pm\theta \in \Delta^\circ$. For $m \geq 2$, it is possible in (4.37) to have $Q_0'(-\theta) = 0$, but $Q_0''(-\theta) > 0$ by (F.9), so $\sigma^2_{\mathrm{IS}(\theta)} \geq mQ_0''(-\theta)[\alpha(\theta)]^m - m^2\mu_0^2$. Also, $\theta \neq 0$ implies that $\alpha(\theta) > 1$ by (F.10). Moreover, we have that $\mu = m\mu_0$. Thus, as $m \to \infty$, the RE of the $\mathrm{IS}(\theta)$ estimator of μ with $\theta \neq 0$ asymptotically grows at rate (ignoring constants) at least $[\alpha(\theta)]^{m/2}/\sqrt{m}$ (and at faster rate $[\alpha(\theta)]^{m/2}$ when $M_0'(-\theta) \neq 0$ because then (F.7) implies $[Q_0'(-\theta)]^2 > 0$ in (4.37)), establishing the RE result in (4.39). Similarly, the WNRE result in (4.39) holds by multiplying the same lower bound for $\mathrm{Var}_{\widetilde{G}_\theta}[c(\mathbf{X})L_\theta(\mathbf{X})]$ by the expected computation time $m\tau_{\mathrm{IS}(\theta)}$ to generate $(c(\mathbf{X}), L_\theta(\mathbf{X}))$ under $\mathbf{X} \sim \widetilde{G}_\theta$.

For $\theta = \theta_\star > 0$ in Assumption 2 as the root of (4.31), it is clear that $\theta_\star$ remains fixed as m grows because the marginal CDF G_0 does not vary with m (Assumption 1) and β is a constant in (4.29). Moreover, (4.39) holds when $\theta = \theta_\star$ because $\theta_\star > 0$, completing the proof of part (ii).

Part (iii): ISDM(θ) We apply arguments from [50]. The denominator in $L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X}) = \frac{\mathrm{d}G(\mathbf{X})}{\delta \mathrm{d}\widetilde{G}_{\theta_\star}(\mathbf{X}) + (1-\delta)\mathrm{d}G(\mathbf{X})}$ is bounded below by $\max[\delta \mathrm{d}\widetilde{G}_{\theta_\star}(\mathbf{X}), (1-\delta)\mathrm{d}G(\mathbf{X})]$, so

$$L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X}) \leq \frac{L_{\theta_\star}(\mathbf{X})}{\delta} \quad \text{and} \quad L_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X}) \leq \frac{1}{1-\delta}. \tag{F.11}$$

Then use a change of measure and exploit the second relation in (F.11) to get

$$\begin{aligned}\sigma^2_{\mathrm{ISDM}(\theta_\star)} &= E_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}\left[c^2(\mathbf{X})L^2_{\mathrm{ISDM}(\theta_\star)}\right] - \mu^2 = E_G\left[c^2(\mathbf{X})L_{\mathrm{ISDM}(\theta_\star)}\right] - \mu^2 \\ &\leq \frac{1}{1-\delta}E_G\left[c^2(\mathbf{X})\right] - \mu^2 = \frac{1}{1-\delta}\left[\sigma^2_{\mathrm{SRS}} + \mu^2\right] - \mu^2 = \frac{\delta\mu_0^2}{1-\delta}m^2 + \frac{\sigma_0^2}{1-\delta}m\end{aligned} \tag{F.12}$$

by (4.35) and because $\mu = m\mu_0$, proving (4.40) (previously shown in [50]), and (4.41) follows.

Part (iv): DE(θ) Note (4.42) holds by (4.35) and (4.37), and (4.43) easily follows, also using (4.34). |

APPENDIX G

PROOF OF THEOREM 7

Part (i): SRS To establish (4.49), which is for $\mathfrak{M} = \text{SRS}$, first recall (4.47) with $\breve{\chi}^2_{\mathfrak{M}} = \breve{\chi}^2_{\text{SRS}} \equiv \chi^2_{\text{SRS}} = p(1-p)$, where $\breve{f}(\breve{\xi})$ evaluates $\breve{f}(x)$ from (4.46) at $x = \breve{\xi}$ from (4.45). Setting $x = \breve{\xi}$ in $mQ_0'(\theta_x) = x$ from (4.46) leads to the relations $mQ_0'(\theta_{\breve{\xi}}) = \breve{\xi} = mQ_0'(\theta_\star)$ by (4.45), so $\theta_{\breve{\xi}} = \theta_\star$. (There cannot be another $\theta' \neq \theta_\star$ in Δ° with $Q_0'(\theta_{\breve{\xi}}) = Q_0'(\theta')$ because (F.9) ensures that $Q_0'(\theta)$ is strictly increasing on Δ°.) We thus obtain

$$\breve{f}(\breve{\xi}) = \frac{1}{\sqrt{2\pi m Q_0''(\theta_\star)}} \exp\Big(mQ_0(\theta_\star) - m\theta_\star Q_0'(\theta_\star)\Big) = \frac{1}{\sqrt{2\pi m Q_0''(\theta_\star)}} e^{-\beta m} \tag{G.1}$$

by (4.31). We then put (4.29) and (G.1) into (4.47) to get

$$\breve{\kappa}^2_{\text{SRS}} = \frac{2\pi m Q_0''(\theta_\star)(1-e^{-\beta m})e^{-\beta m}}{e^{-2\beta m}}, \tag{G.2}$$

verifying (4.49). Also, (4.50) follows from (4.48) for $\mathfrak{M} = \text{SRS}$ and (4.45), completing the proof of (i).

Part (ii): IS($\theta_\star$) Recall $\theta_\star > 0$ from (4.31) in Assumption 2. As shown in [41, Theorem 2], (F.2) implies the numerator $\breve{\chi}^2_{\text{IS}}$ in the approximate asymptotic variance in (4.47) satisfies

$$\begin{aligned}\breve{\chi}^2_{\text{IS}} &\leq E_{\widetilde{G}_{\theta_\star}}\left[L^2_{\theta_\star}(\mathbf{X})\, I(c(\mathbf{X}) > \breve{\xi})\right] = E_{\widetilde{G}_{\theta_\star}}\left[\exp\Big(2\left[-\theta_\star c(\mathbf{X}) + mQ_0(\theta_\star)\right]\Big) I(c(\mathbf{X}) > \breve{\xi})\right] \\ &\leq \exp\Big(2[-\theta_\star \breve{\xi} + mQ_0(\theta_\star)]\Big) = \exp\Big(2[-\theta_\star m Q_0'(\theta_\star) + mQ_0(\theta_\star)]\Big) = \exp(-2\beta m)\end{aligned} \tag{G.3}$$

by (4.45) and (4.31). Then putting (G.3) and (G.1) into (4.47) for $\mathfrak{M} = \text{IS}(\theta_\star)$ verifies (4.51). Substituting (4.51) and (4.45) into (4.48) with $\mathfrak{M} = \text{IS}(\theta_\star)$ finally establish (4.52), completing the proof of (ii).

Part (iii): ISDM$(\theta_\star)$ From the first relation in (F.11) and (F.2), we use the fact that $\theta_\star > 0$ to bound the numerator of the first term in (4.58) as

$$\begin{aligned}
\breve{\chi}^2_{\mathrm{ISDM}(\theta_\star)} &\le E_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}\left[L^2_{\mathrm{ISDM}(\theta_\star)}(\mathbf{X})\, I(c(\mathbf{X}) > \breve{\xi})\right] \\
&\le \frac{1}{\delta^2} E_{\widetilde{G}_{\mathrm{ISDM}(\theta_\star)}}\left[\exp\Big(2\left[-\theta_\star c(\mathbf{X}) + mQ_0(\theta_\star)\right]\Big) I(c(\mathbf{X}) > \breve{\xi})\right] \\
&\le \frac{1}{\delta^2}\exp\Big(2[-\theta_\star \breve{\xi} + mQ_0(\theta_\star)]\Big) = \frac{1}{\delta^2}\exp\Big(2[-\theta_\star m Q_0'(\theta_\star) + mQ_0(\theta_\star)]\Big) \\
&= \frac{1}{\delta^2}\exp(-2\beta m) \qquad \text{(G.4)}
\end{aligned}$$

by (4.45) and (4.31). Combining (G.4) with (G.1) leads to

$$\frac{\breve{\chi}^2_{\mathrm{ISDM}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} \le \frac{2\pi Q_0''(\theta_\star)}{\delta^2} m. \qquad \text{(G.5)}$$

Thus, (4.53) holds, and (4.54) follows.

Part (iv): DE$(\theta_\star)$ Note that (4.55) is a simple consequence of (4.49) and (4.51), and (4.56) easily follows. (For $\mathrm{WNR}\breve{\mathrm{E}}_{\mathrm{DE}(\theta_\star),m}[\xi]$, we use the modification to (4.34) described after (4.48).) |

APPENDIX H

PROOF OF THEOREM 8

Part (i): SRS To establish (4.62), we will separately analyze the three terms in (4.58) for $\mathfrak{M} = \text{SRS}$ to show that as $m \to \infty$, the first term grows at a strictly faster rate than the other two terms, where we recall $\Lambda_{\text{SRS}} = \Lambda^{\dagger}_{\text{SRS}} = \Lambda^{\ddagger}_{\text{SRS}} = 1$. By (4.49), the first term in (4.58) is

$$\frac{\chi^2_{\text{SRS}}}{\breve{f}^2(\breve{\xi})} = [2\pi Q_0''(\theta_\star)]m(1-e^{-\beta m})e^{\beta m} = \Omega(me^{\beta m}) \tag{H.1}$$

as $m \to \infty$, with $\beta > 0$ from (4.29). By (4.35), we write the second term in (4.58) as $\sigma^2_{\text{SRS}} = \sigma_0^2 m = o(me^{\beta m})$ as $m \to \infty$. For the third term in (4.58), we use (4.10) and the Cauchy-Schwarz inequality to get

$$\left|\frac{\gamma_{\text{SRS}}}{\breve{f}(\breve{\xi})}\right| \leq \frac{\chi_{\text{SRS}}}{\breve{f}(\breve{\xi})}\sigma_{\text{SRS}} = \left(\sqrt{2\pi Q_0''(\theta_\star)}\,\sigma_0\right) m\sqrt{(1-e^{-\beta m})}e^{\beta m/2} = o(me^{\beta m}) \tag{H.2}$$

as $m \to \infty$. Thus, the second and third terms in (4.58) are asymptotically negligible compared to the first term, which verifies (4.62). Combining (4.62) with (4.57) establishes (4.63) because the expected computation time to generate $c(\mathbf{X})$ with $\mathbf{X} \sim G$ is $m\tau_{\text{SRS}}$ by Assumption 1, completing the proof of part (i).

Part (ii): IS$(\theta_\star)$ To establish (4.64), we will separately analyze the three terms in (4.58) for $\mathfrak{M} = \text{IS}(\theta_\star)$ to show the second term grows at the strictly fastest rate, where we recall that $\Lambda_{\text{IS}(\theta_\star)} = \Lambda^{\dagger}_{\text{IS}(\theta_\star)} = -\Lambda^{\ddagger}_{\text{IS}(\theta_\star)} = 1$. From (4.37), the second term in (4.58) for $m \geq 2$ satisfies

$$\sigma^2_{\text{IS}(\theta_\star)} = m[\alpha(\theta_\star)]^m\Big(m[Q_0'(-\theta_\star)]^2 + Q_0''(-\theta_\star)\Big) - (m\mu_0)^2 = \Omega(m[\alpha(\theta_\star)]^m) \tag{H.3}$$

as $m \to \infty$ because $\theta_\star > 0$ implies that $\alpha(\theta_\star) > 1$ by Theorem 6(ii) and $Q_0''(-\theta_\star) > 0$ by (4.38). For the first term in (4.58), we have that (4.51) ensures

$$\frac{\breve{\chi}^2_{\text{IS}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} \leq [2\pi Q_0''(\theta_\star)]m = o(m[\alpha(\theta_\star)]^m) \tag{H.4}$$

as $m \to \infty$. For the third term in (4.58), the Cauchy-Schwarz inequality implies

$$\left| -\frac{\breve{\gamma}_{\mathrm{IS}(\theta_\star)}}{\breve{f}(\breve{\xi})} \right| \leq \frac{\breve{\chi}_{\mathrm{IS}(\theta_\star)}}{\breve{f}(\breve{\xi})} \sigma_{\mathrm{IS}(\theta_\star)}$$

$$\leq \left[2\pi Q_0''(\theta_\star) m \left(m[\alpha(\theta_\star)]^m \left(m[Q_0'(-\theta_\star)]^2 + Q_0''(-\theta_\star) \right) - (m\mu_0)^2 \right) \right]^{1/2} \tag{H.5}$$

$$= O(m^{3/2}[\alpha(\theta_\star)]^{m/2}) = o(m[\alpha(\theta_\star)]^m) \tag{H.6}$$

as $m \to \infty$ because $\alpha(\theta_\star) > 1$. Thus, (4.64) holds by (H.3)–(H.6). Combining (4.64) with (4.57) verifies (4.65) as the expected time to generate $(c(\mathbf{X}), L_{\theta_\star}(\mathbf{X}))$ with $\mathbf{X} \sim \widetilde{G}_{\theta_\star}$ is $m\tau_{\mathrm{IS}(\theta_\star)}$, completing the proof of part (ii).

Part (iii): MSIS$(\theta_\star)$ To show (4.66), we separately analyze the first two terms in (4.58) for $\mathfrak{M} = \mathrm{MSIS}(\theta_\star)$ (recall $\Lambda^{\ddagger}_{\mathrm{MSIS}(\theta_\star)} = 0$) to show that each grows asymptotically at most linearly in m. The first term in (4.58) satisfies $\breve{\chi}^2_{\mathrm{IS}}/[\delta \breve{f}^2(\breve{\xi})] \leq [2\pi Q_0''(\theta_\star)/\delta]m$ by (4.51). The second term in (4.58) is $\sigma^2_{\mathrm{SRS}}/(1-\delta) = [\sigma_0^2/(1-\delta)]m$ by (4.35), verifying (4.66). Combining (4.66) with (4.57) establishes the first result (RĔ) in (4.67). The boundedness of WNRĔ in (4.67) holds by putting (4.57), (4.51), and (4.35) into (4.61), completing the proof of part (iii).

Part (iv): ISDM$(\theta_\star)$ To show (4.68), we will separately analyze the three terms in (4.58) for $\mathfrak{M} = \mathrm{ISDM}(\theta_\star)$ to derive an upper bound for each, where we recall $\Lambda_{\mathrm{ISDM}(\theta_\star)} = \Lambda^{\dagger}_{\mathrm{ISDM}(\theta_\star)} = -\Lambda^{\ddagger}_{\mathrm{ISDM}(\theta_\star)} = 1$. The first term in (4.58) satisfies $\frac{\breve{\chi}^2_{\mathrm{ISDM}(\theta_\star)}}{\breve{f}^2(\breve{\xi})} \leq \frac{2\pi Q_0''(\theta_\star)}{\delta^2} m$ by (4.53), and the second term obeys $\sigma^2_{\mathrm{ISDM}(\theta_\star)} \leq \frac{\delta\mu_0^2}{1-\delta} m^2 + \frac{\sigma_0^2}{1-\delta} m$ by (4.40). For the third term in (4.58), the Cauchy-Schwarz inequality implies

$$\left| \frac{-2\breve{\gamma}_{\mathrm{ISDM}(\theta_\star)}}{\breve{f}(\breve{\xi})} \right| \leq 2\frac{\breve{\chi}_{\mathrm{ISDM}(\theta_\star)}}{\breve{f}(\breve{\xi})} \sigma_{\mathrm{ISDM}(\theta_\star)} \leq 2\left[\frac{2\pi Q_0''(\theta_\star)}{\delta^2} m \left(\frac{\delta\mu_0^2}{1-\delta} m^2 + \frac{\sigma_0^2}{1-\delta} m \right) \right]^{1/2}$$

$$= 2\left[\frac{2\pi Q_0''(\theta_\star)}{(1-\delta)} \left(\frac{\mu_0^2}{\delta} + \frac{\sigma_0^2}{m\delta^2} \right) \right]^{1/2} m^{3/2} \tag{H.7}$$

by (G.5) and (F.12). Applying the upper bounds from (G.5), (F.12), and (H.7) in (4.58) verifies (4.68). Combining (4.68) with (4.57) establishes (4.69), completing the proof of part (iv).

Part (v): DE($\theta_\star$), relations (4.70) **and** (4.71) To show (4.70), we will provide lower bounds for each of the two terms in parentheses for $\breve{\zeta}^2_{\mathrm{DE}(\theta_\star)}$ in (4.59). Because $\delta, \upsilon_1, \upsilon_2 \in (0,1)$, the asymptotic rate (in m) at which the first term in (4.59) grows is determined by $\upsilon_2^2\sigma^2_{\mathrm{IS}(\theta_\star)}/\delta$, as in (4.64), so the first term in (4.59) is bounded below by $[\upsilon_2^2 Q_0''(-\theta_\star)/\delta]m[\alpha(\theta_\star)]^m + o(m[\alpha(\theta_\star)]^m)$. Also, as in (4.62) the asymptotic rate at which the second term in (4.59) increases is governed by $\frac{\upsilon_1'^2}{1-\delta}\frac{\chi^2_{\mathrm{SRS}}}{\breve{f}^2(\breve{\xi})}$, so the second term in (4.59) equals $[\frac{\upsilon_1'^2}{1-\delta}2\pi Q_0''(\theta_\star)]m(1-e^{-\beta m})e^{\beta m} + o(m(1-e^{-\beta m})e^{\beta m})$. Combining these two results yields (4.70). Moreover, the first result of (4.71) follows from (4.70) and (4.57). Finally, putting (4.57), (4.49), and (4.37) into (4.60) verifies the second part of (4.71).

Part (v): Equation (4.72) The paragraph before Theorem 8 defines $(\breve{\upsilon}^*_{1,m}, \breve{\upsilon}^*_{2,m}) = \left(\frac{\breve{a}_{1,m}}{\breve{a}_{0,m}}, \frac{\breve{a}_{2,m}}{\breve{a}_{0,m}}\right)$, where

$$\breve{a}_{0,m} = \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} - \breve{\mathrm{C}}^2_{\mathrm{IS},m} + 2\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \breve{\mathrm{C}}^2_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m}, \tag{H.8}$$

$$\breve{a}_{1,m} = \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m}\breve{\mathrm{C}}_{\mathrm{IS},m} + \breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \breve{\mathrm{C}}^2_{\mathrm{SRS},m}, \text{ and} \tag{H.9}$$

$$\breve{a}_{2,m} = \breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} + \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{C}}_{\mathrm{IS},m} + \breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \breve{\mathrm{C}}^2_{\mathrm{SRS},m}. \tag{H.10}$$

We will prove (4.72) by analyzing growth rates of the terms in (H.8)–(H.10) as $m \to \infty$ for fixed $\delta \in (0,1)$. By (4.29) and (F.10), we have $\beta > 0$ and $\alpha(\theta_\star) > 1$. The first term in both (H.8) and (H.9) satisfies

$$\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} = \Omega(m^2[\alpha(\theta_\star)]^m e^{\beta m}) \tag{H.11}$$

as $m \to \infty$, by (H.1) and (H.3). The rest of the proof will show that each other term in (H.8)–(H.10) is $o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m})$ as $m \to \infty$, which will imply (4.72). We start with analyzing the common terms. Note that

$$\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}) \quad \text{and} \quad \breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}), \tag{H.12}$$

where the first (resp., second) relation holds by (H.1) and (H.4) (resp., (4.35) and (H.3)). Then as $m \to \infty$, it follows from (H.2), (H.5), and (H.12) that

$$\breve{\mathrm{C}}^2_{\mathrm{SRS},m} \le \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}) \quad \text{and} \tag{H.13}$$

$$\breve{\mathrm{C}}^2_{\mathrm{IS},m} \le \breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}). \tag{H.14}$$

Now we will repeatedly use (H.11)–(H.14) to establish the growth rates of $\breve{a}_{0,m}$, $\breve{a}_{1,m}$, and $\breve{a}_{2,m}$ as $m \to \infty$. First, (H.8) implies

$$\breve{a}_{0,m} = \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} + o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}). \tag{H.15}$$

To next handle $\breve{a}_{1,m}$ in (H.9), as $m \to \infty$, we get $\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m})$ by (H.1) and (H.2). We also obtain $\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m}\breve{\mathrm{C}}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m})$, because $\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m})$ by (4.35) and (H.1), and $\breve{\mathrm{C}}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m})$ by (H.3) and (H.6). Handling the other terms in (H.9) by (H.11)–(H.13) then leads to

$$\breve{a}_{1,m} = \breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} + o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}) \tag{H.16}$$

as $m \to \infty$. Lastly, for $\breve{a}_{2,m}$ in (H.10), we see that $\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m})$ as $m \to \infty$ because $\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m})$ by (H.3) and (H.4), and $\breve{\mathrm{C}}_{\mathrm{SRS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m})$ by (H.1) and (H.2). Moreover, (H.3) and (H.6) ensure that $\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{C}}_{\mathrm{IS},m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m})$, so as $m \to \infty$,

$$\breve{a}_{2,m} = o(\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}) \tag{H.17}$$

holds by (H.11)–(H.13).

Finally, (4.72) follows from (H.15)–(H.17), completing the proof. |

Next we analyze the approximate optimal value of (υ_1, υ_2) when minimizing $\mathrm{WNR}\breve{\mathrm{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$ in (4.60). In this case, we define the approximation $(\breve{\upsilon}'_{1,m}, \breve{\upsilon}'_{2,m}) = (\frac{\breve{a}'_{1,m}}{\breve{a}'_{0,m}}, \frac{\breve{a}'_{2,m}}{\breve{a}'_{0,m}})$, but now $\breve{a}'_{0,m} = m^2(-\tau^2_{\mathrm{IS}(\theta_\star)}\breve{\mathrm{C}}^2_{\mathrm{IS},m} + 2\tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \tau^2_{\mathrm{SRS}}\breve{\mathrm{C}}^2_{\mathrm{SRS},m} + \tau^2_{\mathrm{IS}(\theta_\star)}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \tau^2_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m})$, $\breve{a}'_{1,m} = m^2(\tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m} + \tau^2_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \tau^2_{\mathrm{SRS}}\breve{\mathrm{C}}^2_{\mathrm{SRS},m})$, and $\breve{a}'_{2,m} = m^2(\tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \tau^2_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m}\breve{\mathrm{V}}^{(\mu)}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{V}}^{(\xi)}_{\mathrm{SRS},m} + \tau_{\mathrm{IS}(\theta_\star)}\tau_{\mathrm{SRS}}\breve{\mathrm{C}}_{\mathrm{IS},m}\breve{\mathrm{C}}_{\mathrm{SRS},m} - \tau^2_{\mathrm{SRS}}\breve{\mathrm{C}}^2_{\mathrm{SRS},m})$. Note that when minimizing $\mathrm{WNR}\breve{\mathrm{E}}_{\mathrm{DE}(\theta_\star),m}[\eta]$, the only difference between $(\breve{\upsilon}'_{1,m}, \breve{\upsilon}'_{2,m})$ and $(\breve{\upsilon}^*_{1,m}, \breve{\upsilon}^*_{2,m})$ in (4.28) is that we further multiplied each product item in $\breve{a}_{0,m}$, $\breve{a}_{1,m}$ and $\breve{a}_{2,m}$ from (H.8)–(H.10) by factors m^2 and a constant involving τ_{SRS} or $\tau_{\mathrm{IS}(\theta)}$ (or both). Thus, an argument analogous to the one using (H.8)–(H.17) to prove (4.72) shows that $\lim_{m\to\infty}(\breve{\upsilon}'_{1,m}, \breve{\upsilon}'_{2,m}) = (1,0)$.

APPENDIX I

TWO-STEP IS TO ESTIMATE EXTREME QUANTILE AND EC IN PCRM WITH RANDOM LOSS GIVEN DEFAULT

We now describe our IS approach to estimate an extreme quantile for the model in Section 4.6.2. We assume that the common shock $S \equiv 1$ in (4.2), as in [40], but we extend their method to allow for the loss given default to be stochastic. Although our simulation experiments in Section 4.6.2 have LGD $C_k \sim \text{Unif}(0, \beta_k)$, independent of $(\mathbf{Z}, \varepsilon_1, \ldots, \varepsilon_m)$, we develop the method for more general LGD satisfying certain conditions; see (I.2). Let V_k denote the marginal CDF (not necessarily uniform) of the LGD C_k.

Reference [40] developed a two-step IS method to estimate the tail probability $\lambda_x \equiv P(Y > x)$, where x is a given large threshold, and we adapt their ideas to estimate the p-quantile ξ, for $p \approx 1$. Their method critically depends on knowing the threshold x, which is explicitly used throughout their approach. We cannot simply apply their technique by letting $x = \xi$, as ξ is unknown. Instead, we first run pilot simulations for a few different values of the threshold x, and interpolate to obtain a crude approximation $\mathring{\xi}$ to ξ. Finally we use additional simulation runs with our modification of the two-step approach of [40] with threshold $x = \mathring{\xi}$, and employ the resulting data to estimate ξ.

Before describing the two-step IS, we first consider a one-step IS conditional on $\mathbf{Z}$, which makes the obligors conditionally independent. In the following, Appendix I.1 first applies the one-step IS conditional on $\mathbf{Z}$ to estimate $\lambda_x(\mathbf{Z}) \equiv P(Y > x \mid \mathbf{Z})$. Next, Appendix I.2 extends the one-step IS to a two-step IS to estimate λ_x, by first using IS to sample $\mathbf{Z}$ from a different CDF than its original one and then applying the one-step IS given the observed $\mathbf{Z}$. Appendix I.3 finally adapts the two-step IS for λ_x to instead estimate ξ.

Recall that the total portfolio loss in Section 4.6.2 is $Y = \sum_{k=1}^{m} C_k D_k$, where $D_k = I(\varepsilon_k > (w_k - \mathbf{a}_k \mathbf{Z})/b_k)$ is the indicator that obligor k defaults because we assumed that the

common shock $S \equiv 1$. The mutual independence of $\mathbf{Z}, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ implies that

$$\text{given } \mathbf{Z}, \text{ the default indicators } D_1, \ldots, D_m \text{ are conditionally independent.} \tag{I.1}$$

Moreover, we assume that for $\mathbf{D} = (D_1, \ldots, D_m)$,

$$C_1, \ldots, C_m, \mathbf{D} \text{ are conditionally independent, given } \mathbf{Z}. \tag{I.2}$$

The following methods will exploit properties (I.1) and (I.2).

I.1 One-Step IS Conditional on Z to Estimate $P(Y > x \mid \mathbf{Z})$

This section will modify the one-step IS of [40] to estimate $\lambda_x(\mathbf{Z})$ when LGD is random and satisfies (I.2). The IS applies exponential twisting (Section 4.5.1). Let $F(\cdot \mid \mathbf{Z})$ be the *conditional CDF* (CCDF) of the loss Y given $\mathbf{Z}$. For each obligor k, define $T_k \equiv C_k D_k$, so $Y = \sum_{k=1}^m T_k$. Let $H_k(\cdot \mid \mathbf{Z})$ be the CCDF of T_k given $\mathbf{Z}$. We will see that given $\mathbf{Z}$, applying an exponential twist to the CCDF $F(\cdot \mid \mathbf{Z})$ of Y with twisting parameter θ is equivalent to twisting each $H_k(\cdot \mid \mathbf{Z})$ with the same θ. (We will later describe in (I.17) how to choose $\theta = \theta_x(\mathbf{Z})$ as a function of both the factor values $\mathbf{Z}$ and the threshold x.)

Exponential Twist to Each $H_k(\cdot \mid \mathbf{Z})$ This section will exponentially twist each $H_k(\cdot \mid \mathbf{Z})$ with the same $\theta \in \mathbb{R}$, and gives details on how to generate T_k when applying IS conditional on $\mathbf{Z}$. The exponential twist $\widetilde{H}_{k,\theta}(\cdot \mid \mathbf{Z})$ of the CCDF $H_k(\cdot \mid \mathbf{Z})$ of T_k given $\mathbf{Z}$ using parameter $\theta \in \mathbb{R}$ is defined by

$$\mathrm{d}\widetilde{H}_{k,\theta}(t \mid \mathbf{Z}) = \frac{e^{\theta t}\,\mathrm{d}H_k(t \mid \mathbf{Z})}{m_{H_k}(\theta, \mathbf{Z})}, \tag{I.3}$$

where $m_{H_k}(\theta, \mathbf{Z}) = \int e^{\theta t}\,\mathrm{d}H_k(t \mid \mathbf{Z})$ is the *conditional moment generating function* (CMGF) of $T_k \sim H_k(\cdot \mid \mathbf{Z})$. Let $\widetilde{E}_\theta[\cdot \mid \mathbf{Z}]$ denote conditional expectation given $\mathbf{Z}$, when each $T_k \sim$

$\widetilde{H}_{k,\theta}(\cdot \mid \mathbf{Z})$. By (I.2) we can write

$$\begin{aligned}\lambda_x(\mathbf{Z}) &= E\left[I\left(\textstyle\sum_{k=1}^m T_k > x\right) \mid \mathbf{Z}\right] = \int_{(t_1,\ldots,t_m)\in\mathbb{R}^m} I\left(\textstyle\sum_{k=1}^m t_k > x\right) \prod_{k=1}^m \mathrm{d}H_k(t_k \mid \mathbf{Z}) \\ &= \int_{(t_1,\ldots,t_m)\in\mathbb{R}^m} I\left(\textstyle\sum_{k=1}^m t_k > x\right) \prod_{k=1}^m \frac{\mathrm{d}H_k(t_k \mid \mathbf{Z})}{\mathrm{d}\widetilde{H}_{k,\theta}(t_k \mid \mathbf{Z})} \mathrm{d}\widetilde{H}_{k,\theta}(t_k \mid \mathbf{Z}) \\ &= \widetilde{E}_\theta[I(\textstyle\sum_{k=1}^m T_k > x) L_\theta(T_1,\ldots,T_m,\mathbf{Z}) \mid \mathbf{Z}], \end{aligned} \tag{I.4}$$

where $L_\theta(T_1,\ldots,T_m,\mathbf{Z}) = \prod_{k=1}^m \mathrm{d}H_k(t_k \mid \mathbf{Z})/\mathrm{d}\widetilde{H}_{k,\theta}(t_k \mid \mathbf{Z})$ is the conditional LR given $\mathbf{Z}$. Also, let $\psi_{H_k}(\theta,\mathbf{Z}) = \ln[m_{H_k}(\theta,\mathbf{Z})]$ be the *conditional cumulant generating function* (CCGF) of $T_k \sim H_k(\cdot \mid \mathbf{Z})$, so by (I.3),

$$L_\theta(T_1,\ldots,T_m,\mathbf{Z}) = \prod_{k=1}^m \frac{\mathrm{d}H_k(t_k \mid \mathbf{Z})}{e^{\theta t_k}\,\mathrm{d}H_k(t_k \mid \mathbf{Z})/m_{H_k}(\theta,\mathbf{Z})} = \prod_{k=1}^m \frac{m_{H_k}(\theta,\mathbf{Z})}{e^{\theta T_k}} = \prod_{k=1}^m e^{\psi_{H_k}(\theta,\mathbf{Z})-\theta T_k}. \tag{I.5}$$

We now give more details on the exponential twist $\widetilde{H}_{k,\theta}(\cdot \mid \mathbf{Z})$ defined by (I.3), which will require expressions for $H_k(\cdot \mid \mathbf{Z})$ and $m_{H_k}(\theta,\mathbf{Z})$. To compute $m_{H_k}(\theta,\mathbf{Z})$, let $p_k(\mathbf{Z})$ be the conditional probability that obligor k defaults given $\mathbf{Z}$, which satisfies

$$p_k(\mathbf{Z}) \equiv P(D_k = 1 \mid \mathbf{Z}) = P\left(\varepsilon_k > \frac{w_k - \mathbf{a}_k\mathbf{Z}}{b_k} \mid \mathbf{Z}\right) = \Phi\left(\frac{\mathbf{a}_k\mathbf{Z} + \Phi^{-1}(p_k)}{b_k}\right) \tag{I.6}$$

because $\varepsilon_k \sim N(0,1)$ is independent of $\mathbf{Z}$ and the $N(0,1)$ density is symmetric about the origin. Then we use (I.2) and (I.6) to compute the CMGF of $T_k = C_k D_k \sim H_k(\cdot \mid \mathbf{Z})$ as

$$\begin{aligned} m_{H_k}(\theta,\mathbf{Z}) &= E\left[e^{\theta C_k D_k} \mid \mathbf{Z}\right] = E\left[E\left[e^{\theta C_k D_k} \mid C_k, \mathbf{Z}\right] \mid \mathbf{Z}\right] \\ &= E\left[e^{\theta C_k \cdot 0}(1 - p_k(\mathbf{Z})) + e^{\theta C_k \cdot 1} p_k(\mathbf{Z}) \mid \mathbf{Z}\right] \\ &= 1 - p_k(\mathbf{Z}) + p_k(\mathbf{Z}) E\left[e^{\theta C_k} \mid \mathbf{Z}\right] = 1 + p_k(\mathbf{Z})[m_{V_k}(\theta,\mathbf{Z}) - 1], \end{aligned} \tag{I.7}$$

where $m_{V_k}(\theta,\mathbf{Z}) = E\left[e^{\theta C_k} \mid \mathbf{Z}\right]$ is the CMGF of $C_k \sim V_k(\cdot \mid \mathbf{Z})$, and $V_k(\cdot \mid \mathbf{Z})$ is the CCDF of C_k given $\mathbf{Z}$.

Next we will work out the details of the conditional exponential twist given $\mathbf{Z}$ for each $T_k \sim H_k(\cdot \mid \mathbf{Z})$. To do this we need an expression for $H_k(\cdot \mid \mathbf{Z})$, which by (I.2) and (I.6)

satisfies

$$\begin{aligned} H_k(t \mid \mathbf{Z}) &= P(C_k D_k \le t \mid \mathbf{Z}) = E[P(C_k D_k \le t \mid D_k, \mathbf{Z}) \mid \mathbf{Z}] \\ &= P(D_k = 0 \mid \mathbf{Z})\, P(C_k D_k \le t \mid D_k = 0, \mathbf{Z}) + P(D_k = 1 \mid \mathbf{Z})\, P(C_k D_k \le t \mid D_k = 1, \mathbf{Z}) \\ &= (1 - p_k(\mathbf{Z}))I(t \ge 0) + p_k(\mathbf{Z}) P(C_k \le t \mid \mathbf{Z}) \end{aligned} \tag{I.8}$$

by (I.6), so $H_k(\cdot \mid \mathbf{Z})$ is a mixture of the CDFs $I(\cdot \ge 0)$ and $V_k(\cdot \mid \mathbf{Z})$. Thus, we have

$$\mathrm{d}H_k(t \mid \mathbf{Z}) = P(T_k \in \mathrm{d}t \mid \mathbf{Z}) = (1 - p_k(\mathbf{Z}))\delta_0(\{\mathrm{d}t\}) + p_k(\mathbf{Z}) v_k(t \mid \mathbf{Z})\, \mathrm{d}t, \tag{I.9}$$

where $v_k(\cdot \mid \mathbf{Z})$ is the density of $V_k(\cdot \mid \mathbf{Z})$, and δ_0 is a measure defined on measurable sets $A \subseteq \mathbb{R}$, such that $\delta_0(A) = 1$ if $0 \in A$ and $\delta_0(A) = 0$ if $0 \notin A$. Then we can write $I(t \ge 0) = \int_{s=-\infty}^{t} \delta_0(\{\mathrm{d}s\}) = \delta_0((-\infty, t])$. Also, putting (I.9) into (I.3) yields

$$\begin{aligned} \mathrm{d}\widetilde{H}_{k,\theta}(t \mid \mathbf{Z}) &= \frac{e^{\theta t}(1 - p_k(\mathbf{Z}))\delta_0(\{\mathrm{d}t\})}{m_{H_k}(\theta, \mathbf{Z})} + \frac{e^{\theta t} p_k(\mathbf{Z}) v_k(t \mid \mathbf{Z})}{m_{H_k}(\theta, \mathbf{Z})}\, \mathrm{d}t \\ &= \frac{(1 - p_k(\mathbf{Z}))\delta_0(\{\mathrm{d}t\})}{m_{H_k}(\theta, \mathbf{Z})} + \frac{e^{\theta t} p_k(\mathbf{Z}) v_k(t \mid \mathbf{Z})}{m_{H_k}(\theta, \mathbf{Z})}\, \mathrm{d}t \qquad \text{(I.10)} \\ &= \widetilde{q}_{k,\theta}(\mathbf{Z})\delta_0(\{\mathrm{d}t\}) + \widetilde{p}_{k,\theta}(\mathbf{Z})\widetilde{v}_{k,\theta}(t \mid \mathbf{Z})\, \mathrm{d}t, \end{aligned}$$

where (I.10) holds because $\delta_0(\{dt\})$ is nonzero only when $t = 0$, in which case $e^{\theta t} = 1$, and

$$\widetilde{v}_{k,\theta}(t \mid \mathbf{Z}) \equiv \frac{e^{\theta t} v_k(t \mid \mathbf{Z})}{m_{V_k}(\theta, \mathbf{Z})},\ \widetilde{q}_{k,\theta}(\mathbf{Z}) \equiv \frac{1 - p_k(\mathbf{Z})}{m_{H_k}(\theta, \mathbf{Z})},\ \widetilde{p}_{k,\theta}(\mathbf{Z}) \equiv 1 - \widetilde{q}_{k,\theta}(\mathbf{Z}) = \frac{p_k(\mathbf{Z}) m_{V_k}(\theta, \mathbf{Z})}{m_{H_k}(\theta, \mathbf{Z})}, \tag{I.11}$$

with (I.11) using (I.7). Note that $\widetilde{v}_{k,\theta}(\cdot \mid \mathbf{Z})$ is the exponential twist of $v_{k,\theta}(\cdot \mid \mathbf{Z})$. Given $\mathbf{Z}$, we have $\widetilde{q}_{k,\theta}(\mathbf{Z}) + \widetilde{p}_{k,\theta}(\mathbf{Z}) = 1$ with $\widetilde{q}_{k,\theta}(\mathbf{Z}) \ge 0$ and $\widetilde{p}_{k,\theta}(\mathbf{Z}) \ge 0$ by (I.7), so

$$\widetilde{H}_{k,\theta}(t \mid \mathbf{Z}) = \int_{s=0}^{t} \mathrm{d}\widetilde{H}_{k,\theta}(s \mid \mathbf{Z}) = \widetilde{q}_{k,\theta}(\mathbf{Z}) I(t \ge 0) + \widetilde{p}_{k,\theta}(\mathbf{Z}) \widetilde{V}_{k,\theta}(t \mid \mathbf{Z}) \tag{I.12}$$

is a mixture of the CDFs $I(t \ge 0)$ and $\widetilde{V}_{k,\theta}(t \mid \mathbf{Z}) \equiv \int_{s=-\infty}^{t} \widetilde{v}_{k,\theta}(s)\, \mathrm{d}s$. Compared to $H_k(\cdot \mid \mathbf{Z})$, the exponential twist $\widetilde{H}_{k,\theta}(\cdot \mid \mathbf{Z})$ shifts the original distribution's mass to the right when $\theta > 0$, making large losses more likely. Also, setting $\theta = 0$ leads to $\widetilde{H}_{k,0}(\cdot \mid \mathbf{Z}) = H_k(\cdot \mid \mathbf{Z})$.

For a given θ, we can generate an observation of $T_k \sim \widetilde{H}_{k,\theta}(\cdot \mid \mathbf{Z})$ as follows. With probability $\widetilde{q}_{k,\theta}(\mathbf{Z})$, we set $T_k = 0$; otherwise (with probability $\widetilde{p}_{k,\theta}(\mathbf{Z})$), we generate $T_k \sim \widetilde{V}_{k,\theta}(\cdot \mid \mathbf{Z})$.

Exponential Twist to $F(\cdot \mid \mathbf{Z})$ We will now show that given $\mathbf{Z}$, exponentially twisting the conditional distribution $F(\cdot \mid \mathbf{Z})$ of $Y = \sum_{k=1}^{m} T_k$ with twisting parameter θ is equivalent to twisting each $H_k(\cdot \mid \mathbf{Z})$ with the same θ. Note that (I.1) and (I.2) imply that the CMGF of Y given $\mathbf{Z}$ satisfies

$$m_F(\theta, \mathbf{Z}) = E\left[\prod_{k=1}^{m} e^{\theta T_k} \mid \mathbf{Z}\right] = \prod_{k=1}^{m} E[e^{\theta T_k} \mid \mathbf{Z}] = \prod_{k=1}^{m} m_{H_k}(\theta, \mathbf{Z}) = \prod_{k=1}^{m} (1 + p_k(\mathbf{Z})[m_{V_k}(\theta, \mathbf{Z}) - 1])$$

by (I.7). Recall that $\psi_{H_k}(\theta, \mathbf{Z})$ is the CCGF of $T_k \sim H_k(\cdot \mid \mathbf{Z})$, so the CCGF of $Y \sim F(\cdot \mid \mathbf{Z})$ is

$$\psi_F(\theta, \mathbf{Z}) = \sum_{k=1}^{m} \psi_{H_k}(\theta, \mathbf{Z}) = \sum_{k=1}^{m} \ln\left(1 + p_k(\mathbf{Z})[m_{V_k}(\theta, \mathbf{Z}) - 1]\right). \tag{I.13}$$

Then by (I.5), we can rewrite the conditional likelihood ratio in (I.4) as

$$L_\theta(T_1, \ldots, T_m, \mathbf{Z}) = e^{\psi_F(\theta, \mathbf{Z}) - \theta Y} \equiv L'_\theta(Y, \mathbf{Z}), \tag{I.14}$$

which depends on $T_1, \ldots, T_m$ through only their sum Y. Hence, given $\mathbf{Z}$, exponentially twisting $F(\cdot \mid \mathbf{Z})$ with θ is equivalent to applying an exponential twist to each $H_k(\cdot \mid \mathbf{Z})$ with the same θ.

Next we will show how we choose the twisting parameter θ given $\mathbf{Z}$. Let $\widetilde{F}_\theta(\cdot \mid \mathbf{Z})$ be the CCDF of Y given $\mathbf{Z}$ under an exponential twist with parameter θ. The conditional expectation $\widetilde{E}_\theta[Y \mid \mathbf{Z}]$ of $Y \sim \widetilde{F}_\theta(\cdot \mid \mathbf{Z})$ under IS given $\mathbf{Z}$ with twisting parameter θ satisfies (e.g., see p. 261 of [38])

$$\widetilde{E}_\theta[Y \mid \mathbf{Z}] = \psi'_F(\theta, \mathbf{Z}) \equiv \frac{\partial}{\partial \theta} \psi_F(\theta, \mathbf{Z}), \tag{I.15}$$

Also, by (I.13), we have that for $m'_{V_k}(\theta, \mathbf{Z}) = \frac{\partial}{\partial \theta} m_{V_k}(\theta, \mathbf{Z})$,

$$\psi'_F(\theta, \mathbf{Z}) = \sum_{k=1}^{m} \frac{p_k(\mathbf{Z}) m'_{V_k}(\theta, \mathbf{Z})}{1 + p_k(\mathbf{Z})(m_{V_k}(\theta, \mathbf{Z}) - 1)}. \tag{I.16}$$

Given $\mathbf{Z}$ and the threshold x in $P(Y > x)$ being estimated, we choose parameter $\theta = \theta_x(\mathbf{Z})$ as follows:

$$\begin{aligned} \text{let } \theta_x(\mathbf{Z}) = 0 \quad &\text{when } x \le \psi'_F(0,\mathbf{Z}); \\ \text{solve for } \theta_x(\mathbf{Z}) \text{ in } \psi'_F(\theta_x(\mathbf{Z}),\mathbf{Z}) = x \quad &\text{when } x > \psi'_F(0,\mathbf{Z}). \end{aligned} \tag{I.17}$$

Here $\psi'_F(\theta_x(\mathbf{Z}),\mathbf{Z}) = \widetilde{E}_{\theta_x(\mathbf{Z})}[Y \,|\, \mathbf{Z}]$ in (I.15), and $\psi'_F(0,\mathbf{Z}) = \widetilde{E}_0[Y \,|\, \mathbf{Z}] = E[Y \,|\, \mathbf{Z}]$, the original conditional mean (without exponential twisting). The conditional event $\{Y > x \,|\, \mathbf{Z}\}$ is typically not rare when $x \le \psi'_F(0,\mathbf{Z})$, so we do not need IS in this case, and (I.17) lets $\theta_x(\mathbf{Z}) = 0$. But when the original conditional mean $\psi'_F(0,\mathbf{Z}) < x$, we choose the twisting parameter $\theta_x(\mathbf{Z})$ in (I.17) so that the conditional mean of Y given $\mathbf{Z}$ under IS equals the threshold x, making the event $\{Y > x \mid \mathbf{Z}\}$ not rare under the IS measure.

I.2 Two-Step IS to Estimate $P(Y > x)$

In this section, we will extend the one-step IS conditional on $\mathbf{Z}$ of Appendix I.1 to estimate the unconditional tail probability λ_x by adapting the two-step IS of [40]. To do this, Appendix I.2 will first specify a new joint CDF $\Gamma_x(\cdot)$ for sampling $\mathbf{Z}$ under IS. Then for a generated $\mathbf{Z} \sim \Gamma_x(\cdot)$, Appendix I.2 will apply the conditional IS from Appendix I.1 on the observed $\mathbf{Z}$, to estimate λ_x.

Specifying the Joint CDF of Z Under IS In this section we will discuss how to choose the new joint CDF for $\mathbf{Z}$ under IS. Let $\Phi_{\mathbf{0}}$ be original joint CDF of vector $\mathbf{Z}$, which has r i.i.d. $N(0,1)$ components, so $\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{z}) = (2\pi)^{-r/2}\exp(-\frac{1}{2}\mathbf{z}^\top\mathbf{z})\,\mathrm{d}\mathbf{z}$. Define the new CDF (not necessarily joint normal) for $\mathbf{Z}$ as $\Gamma_x(\cdot)$, which may depend on the threshold x, satisfying $\mathrm{d}\Gamma_x(\mathbf{z}) > 0$ whenever $\lambda_x(\mathbf{z})\,\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{z}) > 0$. Let $\widetilde{E}_{\Gamma_x}$ be the expectation operator when $\mathbf{Z} \sim \Gamma_x(\cdot)$. Applying a change of measure to $\lambda_x = E\,[\,\lambda_x(\mathbf{Z})\,]$ leads to

$$\lambda_x = \int_{\mathbf{z}\in\mathbb{R}^r} \lambda_x(\mathbf{z})\,\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{z}) = \int_{\mathbf{z}\in\mathbb{R}^r} \lambda_x(\mathbf{z})\frac{\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{z})}{\mathrm{d}\Gamma_x(\mathbf{z})}\,\mathrm{d}\Gamma_x(\mathbf{z}) = \widetilde{E}_{\Gamma_x}\left[\,\lambda_x(\mathbf{Z})\frac{\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{Z})}{\mathrm{d}\Gamma_x(\mathbf{Z})}\,\right]. \tag{I.18}$$

Thus, sampling i.i.d. copies of $\mathbf{Z} \sim \Gamma_x(\cdot)$ and averaging the values of $\lambda_x(\mathbf{Z})\frac{d\Phi_{\mathbf{0}}(\mathbf{Z})}{d\Gamma_x(\mathbf{Z})}$ produces an unbiased estimator of λ_x. Ideally, we would like the optimal choice of $\Gamma_x(\cdot)$ to minimize the variance of the estimator.

Now consider $\Gamma_x^*(\cdot)$ defined by

$$d\Gamma_x^*(\mathbf{z}) \equiv \frac{\lambda_x(\mathbf{z})\, d\Phi_{\mathbf{0}}(\mathbf{z})}{\lambda_x}, \tag{I.19}$$

and $\Gamma_x^*(\cdot)$ is a CDF because $\lambda_x(\mathbf{z}) \geq 0$ and $\int_{\mathbf{z}\in\mathbb{R}^r} d\Gamma_x^*(\mathbf{z}) = 1$ by (I.18). If we let $\Gamma_x(\cdot) = \Gamma_x^*(\cdot)$ in (I.18) and sample $\mathbf{Z} \sim \Gamma_x^*(\cdot)$, then the quantity in the right-hand expectation in (I.18) always satisfies $\lambda_x(\mathbf{Z})\frac{d\Phi_{\mathbf{0}}(\mathbf{Z})}{d\Gamma_x^*(\mathbf{Z})} = \lambda_x$ by (I.19). Hence, the estimator has zero variance, making $\Gamma_x^*(\cdot)$ the optimal (minimum variance) choice of $\Gamma_x(\cdot)$ to estimate λ_x, as is well known (e.g., see p. 256 of [38]). But we cannot implement $\Gamma_x^*(\cdot)$ in practice because it requires knowing λ_x, which is what we want to estimate.

However, $\Gamma_x^*(\cdot)$ defined by (I.19) suggests properties of a "good" choice for $\Gamma_x(\cdot)$. For example, we would like to select $\Gamma_x(\cdot)$ such that $d\Gamma_x(\cdot)$ is large (resp., small) when $\lambda_x(\cdot)\, d\Phi_{\mathbf{0}}(\cdot)$ is large (resp., small). A simple heuristic approach that roughly tries to achieve this chooses the CDF $\Gamma_x(\cdot)$ in (I.18) from within a particular parametric family so that its density has the same mode as $d\Gamma_x^*(\cdot)$. Specifically, we let $\Gamma_x(\cdot) = \Phi_\nu(\cdot)$, which is the joint CDF of r independent normal components, with mean vector $\nu = (\nu_1, \ldots, \nu_r)^\top$ and unit marginal variances. We want to specify ν so that the mode of $d\Phi_\nu(\mathbf{z})$, which is at $\mathbf{z} = \nu$, occurs at the same location as the mode of $d\Gamma_x^*(\mathbf{z})$.

But another issue arises: $\lambda_x(\cdot)$ in (I.19) is also unknown. To handle this, [40] consider replacing $\lambda_x(\cdot)$ with one of several different approximations. We use one of their approaches, which substitutes $\lambda_x(\mathbf{z})$ with the tail probability at threshold x of a (univariate) normal distribution $N(\eta(\mathbf{z}), \sigma_Y^2(\mathbf{z}))$, where $\eta(\mathbf{z}) \equiv E[Y \mid \mathbf{Z} = \mathbf{z}]$ and $\sigma_Y^2(\mathbf{z}) \equiv \mathrm{Var}[Y \mid \mathbf{Z} =$

$\mathbf{z}]$. By (I.1), (I.2), and (I.6), we have

$$\eta(\mathbf{z}) = \sum_{k=1}^{m} E\left[C_k \,|\, \mathbf{Z}=\mathbf{z}\right] E[D_k \,|\, \mathbf{Z}=\mathbf{z}] = \sum_{k=1}^{m} E\left[C_k \,|\, \mathbf{Z}=\mathbf{z}\right] p_k(\mathbf{z}), \quad \text{and}$$
$$\sigma_Y^2(\mathbf{z}) = \sum_{k=1}^{m} \mathrm{Var}[C_k D_k \,|\, \mathbf{Z}=\mathbf{z}] = \sum_{k=1}^{m} \left(E[C_k^2 \,|\, \mathbf{Z}=\mathbf{z}] p_k(\mathbf{z}) - E^2[C_k \mid \mathbf{Z}=\mathbf{z}] p_k^2(\mathbf{z})\right).$$

Thus, we approximate $\lambda_x(\mathbf{z})$ in (I.19) by $\lambda_x^\dagger(\mathbf{z}) \equiv 1 - \Phi\left(\frac{x-\eta(\mathbf{z})}{\sigma_Y(\mathbf{z})}\right)$. The mode-matching heuristic identifies

$$\mathbf{z}_x^\dagger \equiv \underset{\mathbf{z}\in\mathbb{R}^r}{\arg\max} \left[\lambda_x^\dagger(\mathbf{z})\, \mathrm{d}\Phi_{\mathbf{0}}(\mathbf{z})\right] = \underset{\mathbf{z}\in\mathbb{R}^r}{\arg\max} \left[\lambda_x^\dagger(\mathbf{z}) e^{-\mathbf{z}^T\mathbf{z}/2}\right], \tag{I.20}$$

which we can try to compute using numerical optimization methods. (Our simulation experiments employed scipy.optimize with method COBYLA on a few (1 or 2) randomly generated starting points.) Finally, the new joint CDF for $\mathbf{Z}$ under IS is $\Gamma_x(\cdot) = \Phi_\nu(\cdot)$ with mean $\nu = \mathbf{z}_x^\dagger$.

Applying Two-Step IS to Estimate $P(Y > x)$ Now that the joint CDF of $\mathbf{Z}$ under IS has been specified as Φ_ν, this section will extend the one-step IS of Appendix I.1 to a two-step IS to estimate the unconditional tail probability λ_x. We sample $\mathbf{Z}$ under IS from Φ_ν, and $\mathrm{d}\Phi_\nu(\mathbf{z}) = (2\pi)^{-d/2}\exp(-\frac{1}{2}(\mathbf{z}-\nu)^\top(\mathbf{z}-\nu)\,\mathrm{d}\mathbf{z}$. Letting $\Gamma_x(\cdot) = \Phi_\nu(\cdot)$ in (I.18) results in

$$\lambda_x = \widetilde{E}_\nu[\,\lambda_x(\mathbf{Z}) L_\nu^*(\mathbf{Z})\,], \tag{I.21}$$

where $\widetilde{E}_\nu$ is the expectation operator when $\mathbf{Z} \sim \Phi_\nu$, and the likelihood ratio

$$L_\nu^*(\mathbf{Z}) = \frac{\mathrm{d}\Phi_{\mathbf{0}}(\mathbf{Z})}{\mathrm{d}\Phi_\nu(\mathbf{Z})} = \exp\left(\frac{1}{2}\nu^\top\nu - \nu^\top\mathbf{Z}\right) \tag{I.22}$$

corresponds to IS for only $\mathbf{Z}$. Putting (I.4) with $\theta = \theta_x(\mathbf{Z})$ into (I.21) then gives

$$\begin{aligned}
\lambda_x &= \widetilde{E}_\nu\left[\widetilde{E}_{\theta_x(\mathbf{Z})}\left[I(\textstyle\sum_{k=1}^m T_k > x)\, L_{\theta_x(\mathbf{Z})}(T_1,\ldots,T_m,\mathbf{Z}) \mid \mathbf{Z}\right] L_\nu^*(\mathbf{Z})\right] \\
&= \widetilde{E}_\nu\left[\widetilde{E}_{\theta_x(\mathbf{Z})}\left[I(Y>x)\, L'_{\theta_x(\mathbf{Z})}(Y,\mathbf{Z}) L_\nu^*(\mathbf{Z}) \,|\, \mathbf{Z}\right]\right] \\
&= \widetilde{E}_\nu^*\left[I(Y>x)\, L'_{\theta_x(\mathbf{Z})}(Y,\mathbf{Z}) L_\nu^*(\mathbf{Z})\right]
\end{aligned} \tag{I.23}$$

by (I.14) and using iterated expectations, with $\widetilde{E}^*_\nu$ as the expectation corresponding to two-step IS, where we first generate $\mathbf{Z} \sim \Phi_\nu$, and then given $\mathbf{Z}$, we generate Y from the conditional distribution $\widetilde{F}_{\theta_x(\mathbf{Z})}(\cdot \mid \mathbf{Z})$ with twisting parameter $\theta_x(\mathbf{Z})$ in (I.17). Thus, the right side of (I.23) shows that using this two-step IS approach leads to $I(Y > x) \cdot L'_{\theta_x(\mathbf{Z})}(Y, \mathbf{Z}) \cdot L^*_\nu(\mathbf{Z})$ as an unbiased estimator of $P(Y > x)$ based on a single run.

We now detail the two-step IS to estimate λ_x with multiple runs. We first

0. Compute the mean ν for the CDF Φ_ν of $\mathbf{Z}$ under IS as $\nu = \mathbf{z}^\dagger_x$ from (I.20).

We execute step 0 only once. For a sample size n, do the following in each run $i = 1, 2, \ldots, n$:

1. Generate $\mathbf{Z}_i \sim \Phi_\nu$.

2. Compute the twisting parameter $\theta_i = \theta_x(\mathbf{Z}_i)$ using (I.17), for $\psi'_F(0, \mathbf{Z}_i)$ in (I.16).

3. Given $\mathbf{Z}_i$, for each obligor $k = 1, 2, \ldots, m$, if $\theta_i = 0$, generate $T_{k,i}$ from $H_k(\cdot \mid \mathbf{Z}_i)$ in (I.8); else (when $\theta_i > 0$), generate $T_{k,i}$ from $\widetilde{H}_{k,\theta_i}(\cdot \mid \mathbf{Z}_i)$ in (I.12).

4. Compute $Y_i = \sum_{k=1}^m T_{k,i}$, which has CCDF $\widetilde{F}_{\theta_i}(\cdot \mid \mathbf{Z}_i)$.

5. Compute $L'_{\theta_i}(Y_i, \mathbf{Z}_i)$ and $L^*_\nu(\mathbf{Z}_i)$ using (I.14) and (I.22).

After completing all n runs, we obtain an unbiased estimator of λ_x as

$$\widehat{\lambda}^*_n \equiv \frac{1}{n} \sum_{i=1}^n I(Y_i > x) L'_{\theta_i}(Y_i, \mathbf{Z}_i) L^*_\nu(\mathbf{Z}_i).$$

I.3 Two-step IS to Estimate Quantile

Now we adapt the two-step IS method for estimating the unconditional tail probability $\lambda_x = P(Y > x)$ to instead estimate the p-quantile ξ, which equals the threshold x satisfying $P(Y > x) = 1 - p$. The two-step IS approach of Appendix I.2 to estimate λ_x for some fixed x critically depends on knowing the threshold x. As the p-quantile ξ is unknown, we cannot directly apply this IS method with $x = \xi$ to estimate ξ. Instead, we run pilot simulations (with small sample size n_0) with the two-step IS method at a small number j_0 of thresholds

x_j, $j = 1, 2, \ldots, j_0$, estimating the tail probability λ_{x_j} for each j, and then interpolate to obtain a crude approximation $\mathring{\xi}$ to the quantile. Then we run additional simulations using the two-step IS approach of Appendix I.2 for estimating λ_x with $x = \mathring{\xi}$, and use the resulting data to estimate ξ.

In our experiment, we implemented the approximation method for the p-quantile via three steps:

1. Let $x_j = (1 - \alpha_p^j) y^*$ for $j = 1, \ldots, j_0$, where $0 < \alpha_p < 1$ is a constant that may depend on p and other model parameters, and y^* is the maximum possible loss, which is assumed known. Our simulation experiments use $\alpha_p = 0.95$ and $j_0 = 5$. Also, $y^* = \sum_{k=1}^m \beta_k$ as our experiments have the LGD $C_k \sim \text{Unif}(0, \beta_k)$.

2. For each $j = 1, \ldots, j_0$, use x_j as the threshold in the two-step IS algorithm of Appendix I.2 with sample size n_0 to obtain an estimate $\widehat{\lambda}_{x_j}$ of the tail probability $\lambda_{x_j} = P(Y > x_j)$. We let $n_0 = 100$ in our simulation experiments.

3. Find the $j^* \in \{1, 2, \ldots, j_0 - 1\}$ such that $\widehat{\lambda}_{x_{j^*}} \le 1 - p < \widehat{\lambda}_{x_{j^*+1}}$, and use log-interpolation on $(x_{j_*}, \widehat{\lambda}_{x_{j^*}})$ and $(x_{j_*+1}, \widehat{\lambda}_{x_{j^*+1}})$ to obtain $\mathring{\xi}$ as our p-quantile approximation. If $1 - p$ is not between any pair of the $\widehat{\lambda}_{x_j}$, we may need to alter α_p to end up with $\widehat{\lambda}_{x_{j^*}} \le 1 - p < \widehat{\lambda}_{x_{j^*+1}}$ for some j^*.

After obtaining the quantile approximation $\mathring{\xi}$, to implement the two-step IS to estimate ξ, apply steps 0–5 of the algorithm in Appendix I.2 with threshold $x = \mathring{\xi}$ and sample size n, resulting in outputs $(Y_i, \mathbf{Z}_i, \theta_i)$, $i = 1, 2, \ldots, n$. Then we can compute the IS p-quantile estimator via the algorithm described after (4.14) as follows. Let $Y_{1:n} \le Y_{2:n} \le \cdots \le Y_{n:n}$ be the sorted values of $Y_1, Y_2, \ldots, Y_n$. Also, let $L_{i::n} = L'_{\theta_j}(Y_j, \mathbf{Z}_j) L^*_{\nu}(\mathbf{Z}_j)$ for $(Y_j, \mathbf{Z}_j, \theta_j)$ corresponding to $Y_{i:n}$. Finally, our IS p-quantile estimator is $\widehat{\xi}_{\text{IS},n} = Y_{i_p:n}$, where i_p is the greatest integer for which $\sum_{\ell=i_p}^n L_{i::n} \ge n(1-p)$.

REFERENCES

[1] A. Agrawal, G. Fohler, J. Freitag, J. Nowotsch, S. Uhrig, and M. Paulitsch. Contention-aware dynamic memory bandwidth isolation with predictability in COTS multicores: An avionics case study. In *Euromicro Conference on Real-Time Systems (ECRTS)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[2] A. Alban, H. Darji, A. Imamura, and M. K. Nakayama. Efficient Monte Carlo methods for estimating failure probabilities. *Reliability Engineering and System Safety*, 165:376–394, 2017.

[3] A. Alhammad and R. Pellizzoni. Trading cores for memory bandwidth in real-time systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–11. Piscataway, NJ: IEEE, 2016.

[4] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Spring Joint Computer Conference*, pages 483–485, 1967.

[5] L. Andersen and J. Sidenius. Extensions to the Gaussian copula: Random recovery and random factor loadings. *Journal of Credit Risk*, 1(1):29–70, 2005.

[6] B. Andersson and D. de Niz. Analyzing Global-EDF for multiprocessor scheduling of parallel tasks. In *International Conference On Principles Of Distributed Systems*, pages 16–30. New York, NY: Springer, 2012.

[7] ARM. Memory system resource partitioning and monitoring (MPAM), 2018. Retrieved on June, 2022: `https://developer.arm.com/documentation/ddi0598/latest/`.

[8] S. Asmussen. Conditional Monte Carlo for sums, with applications to insurance and finance. *Annals of Actuarial Science*, 12(2):455–478, 2018.

[9] S. Asmussen and P. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, New York City, NY, 2007.

[10] A. N. Avramidis and J. R. Wilson. Correlation-induction techniques for estimating quantiles in simulation. *Operations Research*, 46:574–591, 1998.

[11] M. A. Awan, K. Bletsas, P. F. Souto, B. Akesson, and E.Tovar. Mixed-criticality scheduling with dynamic redistribution of shared cache. In *Euromicro Conference on Real-Time Systems (ECRTS)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[12] R. R. Bahadur. A note on quantiles in large samples. *Annals of Mathematical Statistics*, 37(3):577–580, 1966.

[13] S. Baruah. Federated scheduling of sporadic DAG task systems. In *International Parallel and Distributed Processing Symposium (IPDPS)*, pages 179–186. Piscataway, NJ: IEEE, 2015.

[14] S. Baruah. The federated scheduling of systems of mixed-criticality sporadic dag tasks. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 227–236, 2016.

[15] A. Bassamboo, S. Juneja, and A. Zeevi. Portfolio credit risk with extremal dependence: Asymptotic analysis and efficient simulation. *Operations Research*, 56(3):593–606, 2008.

[16] C. Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, 2011. Retrieved on June, 2022: `http://parsec.cs.princeton.edu`.

[17] P. Billingsley. *Probability and Measure*. New York, NY: John Wiley and Sons, 3rd edition, 1995.

[18] D. A. Bloch and J. L. Gastwirth. On a simple estimate of the reciprocal of the density function. *Annals of Mathematical Statistics*, 39:1083–1085, 1968.

[19] V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, and Andreas Wiese. Feasibility analysis in the sporadic DAG task model. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 225–233, 2013.

[20] G. Chen, B. Hu, K. Huang, A. Knoll, D. Liu, and T. Stefanov. Automatic cache partitioning and time-triggered scheduling for real-time MPSoCs. In *International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, pages 1–8. Piscataway, NJ: IEEE, 2014.

[21] F. Chu and M. K. Nakayama. Confidence intervals for quantiles when applying variance-reduction techniques. *ACM Transactions on Modeling and Computer Simulation*, 22(2):10:1–10:25, 2012.

[22] H. S. Chwa, J. Lee, K. Phan, A. Easwaran, and I. Shin. Global EDF schedulability analysis for synchronous parallel tasks on multicore platforms. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 25–34, 2013.

[23] R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914, 1976.

[24] Deutsche Bank. Annual report 2018. Frankfurt am Main, Germany, 2018.

[25] J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, Cambridge, U.K., 2010.

[26] H. Dong and M. K. Nakayama. Quantile estimation using conditional Monte Carlo and Latin hypercube sampling. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, editors, *Winter Simulation Conference*, pages 1986–1997, Piscataway, New Jersey, 2017. Institute of Electrical and Electronics Engineers.

[27] J. Dong, M. B. Feng, and B. L. Nelson. Unbiased metamodeling via likelihood ratios. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, editors, *Winter Simulation Conference*, pages 1778–1789, Piscataway, New Jersey, 2018. Institute of Electrical and Electronics Engineers.

[28] D. A. Dube, R. R. Sherry, J. R. Gabor, and S. M. Hess. Application of risk informed safety margin characterization to extended power uprate analysis. *Reliability Engineering and System Safety*, 129:19–28, 2014.

[29] R. Durrett. *Probability: Theory and Examples*. Duxbury Press, Belmont, California, 2nd edition, 1996.

[30] S. Farinelli and M. Shkolnikov. Two models of stochastic loss given default. *Journal of Credit Risk*, 8(2):3–20, 2012.

[31] M. Feng and J. Staum. Green simulation: reusing the output of repeated experiments. *ACM Transactions on Modeling and Computer Simulation*, 27(4):23:1–23:28, 2017.

[32] T. Ferguson. Asymptotic joint distribution of sample mean and a sample quantile. Retrieved on June, 2022: `https://www.math.ucla.edu/~tom/papers/unpublished/meanmed.pdf`, 1999.

[33] M. C. Fu, L. J. Hong, and J.-Q. Hu. Conditional Monte Carlo estimation of quantile sensitivities. *Management Science*, 55:2019–2027, 2009.

[34] M. C. Fu and J. Q. Hu. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. New York, NY: Kluwer Academic, 1997.

[35] L. Funaro, O. A. Ben-Yehuda, and A. Schuster. Ginseng: Market-driven LLC allocation. In *USENIX Annual Technical Conference (ATC)*, pages 295–308, 2016.

[36] G. Gamrath, D. Anderson, K. Bestuzheva, W. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, et al. The SCIP Optimization Suite 7.0. In *Technical Report*. 2020.

[37] J. K. Ghosh. A new proof of the Bahadur representation of quantiles and an application. *Annals of Mathematical Statistics*, 42:1957–1961, 1971.

[38] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. New York, NY: Springer, 2004.

[39] P. Glasserman and S. Juneja. Uniformly efficient importance sampling for the tail distribution of sums of random variables. *Mathematics of Operations Research*, 33(1):36–50, 2008.

[40] P. Glasserman and J. Li. Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656, 2005.

[41] P. W. Glynn. Importance sampling for Monte Carlo estimation of quantiles. In S. M. Ermakov and V. B. Melas, editors, *Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation*, pages 180–185, St. Petersburg, Russia, 1996. Publishing House of St. Petersburg Univ.

[42] A. Goyal, P. Shahabuddin, P. Heidelberger, V. Nicola, and P. W. Glynn. A unified framework for simulating Markovian models of highly dependable systems. *IEEE Transactions on Computers*, C-41(1):36–51, 1992.

[43] G. Gracioli, A. Alhammad, R. Mancuso, A. A. Fröhlich, and R. Pellizzoni. A survey on cache management mechanisms for real-time embedded systems. *Computing Surveys (CSUR)*, 48(2):1–36, 2015.

[44] N. Guan, M. Stigge, W. Yi, and G. Yu. Cache-aware scheduling and analysis for multicores. In *International conference on Embedded software (EMSOFT)*, pages 245–254. ACM, 2009.

[45] D. Guo, M. Hassan, R. Pellizzoni, and H. Patel. A comparative study of predictable DRAM controllers. *Transactions on Embedded Computing Systems (TECS)*, 17(2):1–23, 2018.

[46] M. Hassan, H. Patel, and R. Pellizzoni. A framework for scheduling DRAM memory accesses for multi-core mixed-time critical systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 307–316. IEEE, 2015.

[47] Z. He. On the error rate of conditional quasi-Monte Carlo for discontinuous functions. *SIAM Journal on Numerical Analysis*, 57(2):854–874, 2019.

[48] Z. He and X. Wang. Convergence of randomized quasi-Monte Carlo sampling for quantile and expected shortfall. arXiv:1706.00540, 2017.

[49] S. Hess, J. Gaertner, J. Gabor, L. Shanley, L. Enos-Sylla, S. Prasad, and S. Hollingsworth. Framework for risk-informed safety margin characterization. Technical Report 1019206, Electric Power Research Institute, Palo Alto, California, 2009.

[50] T. Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

[51] Intel. Intel CilkPlus v1.2, Sep 2013. Retrieved on June, 2022: `https://www.cilkplus.org/sites/default/files/open_specifications/Intel_Cilk_plus_lang_spec_1.2.htm`.

[52] Intel. User space software for intel(r) resource director technology, 2019. Retrieved on June, 2022: `https://github.com/intel/intel-cmt-cat`.

[53] J. L. Jensen. *Saddlepoint Approximations*. Oxford University Press, New York, 1995.

[54] X. Jiang, N. Guan, X. Long, and W. Yi. Semi-federated scheduling of parallel real-time tasks on multiprocessors. In *Real-Time Systems Symposium (RTSS)*, pages 80–91. IEEE, 2017.

[55] X. Jiang, X. Long, N. Guan, and H. Wan. On the decomposition-based Global EDF scheduling of parallel real-time tasks. In *Real-Time Systems Symposium (RTSS)*, pages 237–246. IEEE, 2016.

[56] X. Jin and A. X. Zhang. Reclaiming quasi-Monte Carlo efficiency in portfolio value-at-risk simulation through Fourier transform. *Management Science*, 52(6):925–938, 2006.

[57] P. Jorion. *Value at Risk*. McGraw-Hill, New York, 3rd edition, 2007.

[58] P. Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, New York, 3rd edition, 2007.

[59] P. Jorion. *Financial Risk Manager Handbook*. John Wiley and Sons, Inc., Hoboken, New Jersey, 6th edition, 2011.

[60] Z. T. Kaplan, Y. Li, and M. K. Nakayama. Monte Carlo estimation of economic capital. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, editors, *Winter Simulation Conference*, pages 1754–1765, Piscataway, New Jersey, 2018. Institute of Electrical and Electronics Engineers.

[61] Z. T. Kaplan, Y. Li, M. K. Nakayama, and B. Tuffin. Randomized quasi-Monte Carlo for quantile estimation. In N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, editors, *Winter Simulation Conference*, pages 428–439, Piscataway, New Jersey, 2019. Institute of Electrical and Electronics Engineers.

[62] J. Kiefer. On Bahadur's representation of sample quantiles. *Annals of Mathemetical Statistics*, 38:1350–1353, 1967.

[63] H. Kim, D. D. Niz, B. Andersson, M. Klein, O. Mutlu, and R. Rajkumar. Bounding memory interference delay in COTS-based multi-core systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 145–154. IEEE, 2014.

[64] H. Kim and R. Rajkumar. Real-time cache management for multi-core virtualization. In *International Conference on Embedded Software (EMSOFT)*, pages 1–10. IEEE, 2016.

[65] J. Kim, H. Kim, K. Lakshmanan, and R. R. Rajkumar. Parallel scheduling for cyber-physical systems: Analysis and case study on a self-driving car. In *4th International Conference on Cyber-Physical Systems (ICCPS)*, pages 31–40, 2013.

[66] P. Klaassen and I. van Eeghen. *Economic Capital: How It Works, and What Every Manager Needs to Know*. Elsevier, Burlington, MA, 2009.

[67] F. Y. Kuo and D. Nuyens. Application of quasi-monte carlo methods to elliptic PDEs with random diffusion coefficients – a survey of analysis and implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696, 2016.

[68] Y. Lai and K.S. Tan. Simulation of nonlinear portfolio value-at-risk by Monte Carlo and quasi-Monte Carlo methods. In M. Holder, editor, *Financial Engineering and Applications*, pages 65–66, Calgary, 2006. ACTA Academic Press.

[69] K. Lakshmanan, S. Kato, and R. Rajkumar. Scheduling parallel real-time tasks on multi-core processors. In *31st IEEE Real-Time Systems Symposium (RTSS)*, pages 259–268, 2010.

[70] P. L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.

[71] P. L'Ecuyer. Randomized quasi-Monte Carlo: An introduction for practitioners. In P. W. Glynn and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC 2016*, pages 29–52, Berlin, 2018. Springer-Verlag.

[72] P. L'Ecuyer, J. H. Blanchet, B. Tuffin, and P. W. Glynn. Asymptotic robustness of estimators in rare-event simulation. *ACM Transactions on Modeling and Computer Simulation*, 20(1):Article 6, 2010.

[73] P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.

[74] P. L'Ecuyer and C. Lemieux. Recent advances in randomized quasi-Monte Carlo methods. In M. Dror, P. L'Ecuyer, and F. Szidarovszki, editors, *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, pages 419–474. Kluwer Academic Publishers, Norwell, 2002.

[75] P. L'Ecuyer and D. Munger. Algorithm 958: Lattice builder: A general software tool for constructing rank-1 lattice rules. *ACM Transactions on Mathematical Software*, 42(2):Article 15, 2016.

[76] P. L'Ecuyer, D. Munger, and B. Tuffin. On the distribution of integration error by randomly-shifted lattice rules. *Electronic Journal of Statistics*, 4:950–993, 2010.

[77] P. L'Ecuyer and G. Perron. On the convergence rates of IPA and FDC derivative estimators for finite-horizon stochastic simulations. *Operations Research*, 42(4):643–656, 1994.

[78] P. L'Ecuyer, F. Puchhammer, and A. Ben Abdellah. Monte Carlo and quasi-Monte Carlo density estimation via conditioning. 2020.

[79] C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Series in Statistics. Springer, New York, 2009.

[80] C. Lemieux and P. L'Ecuyer. Selection criteria for lattice rules and other low-discrepancy point sets. *Mathematics and Computers in Simulation*, 55(1–3):139–148, 2001.

[81] J. Li, K. Agrawal, C. Lu, and C. Gill. Analysis of global edf for parallel tasks. In *25th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 3–13, 2013.

[82] J. Li, Jian-Jia Chen, K. Agrawal, C.Lu, C.D. Gill, and Abusayeed Saifullah. Analysis of federated and global scheduling for parallel real-time tasks. In *26th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 85–96, 2014.

[83] J. Li, S. Dinh, K. Kieselbach, K. Agrawal, K. Gill, and C. Lu. Randomized work stealing for large scale soft real-time systems. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 203–214, 2016.

[84] Y. Li, B. Akesson, and K. Goossens. Architecture and analysis of a dynamically-scheduled real-time memory controller. *Real-Time Systems*, 52(5):675–729, 2016.

[85] P.-E. Lin, K.-T. Wu, and I. A. Ahmad. Asymptotic joint distribution of sample quantiles and sample mean with applications. *Communications in Statistics - Theory and Methods*, A9(1):51–60, 1980.

[86] T. Liu and E. Zhou. Simulation optimization by reusing past replications: don't be afraid of dependence. In K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, editors, *Winter Simulation Conference*, pages 2923–2934, Piscataway, New Jersey, 2020. Institute of Electrical and Electronics Engineers.

[87] W.-L. Loh. On the asymptotic distribution of scrambled net quadrature. *Annals of Statistics*, 31(4):1282–1324, 2003.

[88] E. Lütkebohmert. *Concentration Risk in Credit Portfolios*. Springer, Berlin, 2009.

[89] J. D. McCalpin. Memory bandwidth and machine balance in current high performance computers. *Computer society technical committee on computer architecture (TCCA) newsletter*, 2(19-25), 1995.

[90] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[91] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, Tools*. Princeton University Press, Princeton, New Jersey, revised edition, 2015.

[92] A. Melani, M. Bertogna, V. Bonifaci, A. Marchetti-Spaccamela, and G. Buttazzo. Schedulability analysis of conditional parallel task graphs in multicore systems. *IEEE Transactions on Computers*, 66(2):339–353, 2016.

[93] C. H. Nagaraja and H. N. Nagaraja. Distribution-free approximate methods for constructing confidence intervals for quantiles. *International Statistical Review*, 88(1):75–100, 2020.

[94] M. K. Nakayama. Confidence intervals using sectioning for quantiles when applying variance-reduction techniques. *ACM Transactions on Modeling and Computer Simulation*, 24(4):19:1–19:21, 2014.

[95] M. K. Nakayama. Quantile estimation when applying conditional Monte Carlo. In Mohammad S. Obaidat, Janusz Kacprzyk, and Tuncer Ören, editors, *Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, pages 280–285, Piscataway, New Jersey, 2014. Institute of Electrical and Elect.

[96] M. K. Nakayama. Estimating a failure probability using a combination of variance-reduction techniques. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, editors, *Winter Simulation Conference*, pages 621–632, Piscataway, New Jersey, 2015. Institute of Electrical and Electronics Engineers.

[97] G. Nelissen, V. Berten, J. Goossens, and D. Milojevic. Techniques optimizing the number of processors to schedule multi-threaded tasks. In *24th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 321–330, 2012.

[98] V. A. Nguyen, D. Hardy, and I. Puaut. Cache-conscious off-line real-time scheduling for multi-core platforms: algorithms and implementation. *Real-Time Systems*, 55(4):810–849, 2019.

[99] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63. SIAM, Philadelphia, 1992.

[100] G. Ökten, B. Tuffin, and V. Burago. A central limit theorem and improved error bounds for a hybrid-Monte Carlo sequence with applications in computational finance. *Journal of Complexity*, 22(4):435–458, 2006.

[101] OpenMP. OpenMP Application Program Interface v4.0, July 2013. Retrieved on June, 2022: `http://http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf`.

[102] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, Philadelphia, 2000.

[103] A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[104] A. B. Owen. Randomly permuted (t,m,s)-nets and (t,s)-sequences. In Harald Niederreiter and Peter Jay-Shyong Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing: Lecture Notes in Statistics*, volume 106, pages 299–317. Springer, New York, 1995.

[105] A. B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM Journal of Numerical Analysis*, 34:1884–1910, 1997.

[106] A. B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562, 1997.

[107] A. B. Owen. Scrambling Sobol' and Niedeerreiter-Xing points. *Journal of Complexity*, 14(4):466–489, 1998.

[108] A. B. Owen. *Monte Carlo Theory, Methods and Examples*. Draft, 2019. In preparation.

[109] A. Papageorgiou and S. H. Paskov. Deterministic simulation for risk management. *Journal of Portfolio Management*, 25(5):122–127, 1999.

[110] S. Paskov and J. Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, 22:113–120, 1995.

[111] PBBS. Problem based benchmark suite, 2014. `http://www.cs.cmu.edu/~pbbs`.

[112] R. Pellizzoni and H. Yun. Memory servers for multicore systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–12. IEEE, 2016.

[113] A. Saifullah, J. Li, K. Agrawal, C. Lu, and C. Gill. Multi-core real-time scheduling for generalized parallel task models. *Real-Time Systems*, 49(4):404–435, 2013.

[114] A. Sarkar, F. Mueller, and H. Ramaprasad. Static task partitioning for locked caches in multicore real-time systems. *Transactions on Embedded Computing Systems (TECS)*, 14(1):1–30, 2015.

[115] M. Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[116] S. Scandizzo. *The Validation of Risk Models: A Handbook for Practitioners*. Palgrave Macmillan, New York, 2016.

[117] R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley and Sons, New York, 1980.

[118] R. R. Sherry, J. R. Gabor, and S. M. Hess. Pilot application of risk informed safety margin characterization to a total loss of feedwater event. *Reliability Engineering and System Safety*, 117:65–72, 2013.

[119] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *Internet of Things Journal*, 3(5):637–646, 2016.

[120] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Carendon Press, Oxford, UK, 1994.

[121] P. Sohal, R. Tabish, U. Drepper, and R. Mancuso. E-WarP: A system-wide framework for memory bandwidth profiling and management. In *Real-Time Systems Symposium (RTSS)*, pages 345–357. IEEE, 2020.

[122] L. Sun and L. J. Hong. Asymptotic representations for importance-sampling estimators of value-at-risk and conditional value-at-risk. *Operations Research Letters*, 38(4):246–251, 2010.

[123] C. Tessler, V. Modekurthy P, N. Fisher, and A. Saifullah. Bringing inter-thread cache benefits to federated scheduling. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 281–295. IEEE, 2020.

[124] B. Tuffin. On the use of low discrepancy sequences in Monte Carlo methods. *Monte Carlo Methods and Applications*, 2(4):295–320, 1996.

[125] B. Tuffin. Variance reduction order using good lattice points in Monte Carlo methods. *Computing*, 61(4):371–378, 1998.

[126] B. Tuffin. Randomization of quasi-monte carlo methods for error estimation: Survey and normal approximation. *Monte Carlo Methods and Applications*, 10(3-4):617–628, 2004.

[127] N. Ueter, G. von der Bruggen, J. Chen, J. Li, and K. Agrawal. Reservation-based federated scheduling for parallel real-time tasks. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 482–494, 2018.

[128] U.S. Nuclear Regulatory Commission. Final safety evaluation for WCAP-16009-P, revision 0, 'realistic large break LOCA evaluation methodology using automated statistical treatment of uncertainty method (ASTRUM)' (TAC no. MB9483). Technical report, U.S. Nuclear Regulatory Commission, Washington, DC, 2005.

[129] U.S. Nuclear Regulatory Commission. Final safety evaluation for WCAP-16009-P, revision 0, 'realistic large break LOCA evaluation methodology using automated statistical treatment of uncertainty method (ASTRUM)' (TAC no. MB9483). Technical report, Washington, DC, 2005.

[130] U.S. Nuclear Regulatory Commission. Acceptance criteria for emergency core cooling systems for light-water nuclear power reactors. Title 10, Code of Federal Regulations Section 50.46 (10CFR50.46), U.S. Nuclear Regulatory Commission, Washington, DC, 2010.

[131] U.S. Nuclear Regulatory Commission. Applying statistics. U.S. Nuclear Regulatory Commission Report NUREG-1475, Rev 1, U.S. Nuclear Regulatory Commission, Washington, DC, 2011.

[132] P. K. Valsan, H. Yun, and F. Farshchi. Taming non-blocking caches to improve isolation in multicore real-time systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–12. IEEE, 2016.

[133] Q. Wang and G. Parmer. Fjos: Practical, predictable, and efficient system support for fork/join parallelism. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE 20th*, pages 25–36, 2014.

[134] W. Whitt. *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues*. Springer-Verlag, New York, 2002.

[135] J. Xiao, S. Altmeyer, and A. Pimentel. Schedulability analysis of non-preemptive real-time scheduling for multicore processors with shared caches. In *Real-Time Systems Symposium (RTSS)*, pages 199–208. IEEE, 2017.

[136] M. Xu, LTX. Phan, H. Choi, Y. Lin, H. Li, C. Lu, and I. Lee. Holistic resource allocation for multicore real-time systems. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 345–356. IEEE, 2019.

[137] Y. Ye, R. West, J. Zhang, and Z. Cheng. Maracas: A real-time multicore vCPU scheduling framework. In *Real-Time Systems Symposium (RTSS)*, pages 179–190. IEEE, 2016.

[138] H. Yun, W. Ali, S. Gondi, and S. Biswas. Bwlock: A dynamic memory access control framework for soft real-time applications on multicore platforms. *Transactions on Computers (TC)*, 66(7):1247–1252, 2016.

[139] H. Yun, R. Mancuso, Z. Wu, and R. Pellizzoni. Palloc: Dram bank-aware memory allocator for performance isolation on multicore platforms. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 155–166. IEEE, 2014.

[140] H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha. Memguard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 55–64. IEEE, 2013.

[141] Heechul Yun, Gang Yao, Rodolfo Pellizzoni, Marco Caccamo, and Lui Sha. Memory bandwidth management for efficient performance isolation in multi-core platforms. *Transactions on Computers (TC)*, 65(2):562–576, 2015.

[142] X. Zhang, S. Dwarkadas, and K. Shen. Towards practical page coloring-based multicore cache management. In *European conference on Computer systems*, pages 89–102. ACM, 2009.

[143] Y. Zhou and D. Wentzlaff. Mitts: Memory inter-arrival time traffic shaping. *SIGARCH Computer Architecture News*, 44(3):532–544, 2016.

[144] A. Zuepke and R. Kaiser. Deterministic futexes: Addressing wcet and bounded interference concerns. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 65–76. IEEE, 2019.