Title

Efficient and Trustworthy Algorithm Design for Large-Scale Simulation and Inference

Permalink

https://escholarship.org/uc/item/4mt962h7

Author

Xu, Jingxu

Publication Date

2024

Peer reviewed|Thesis/dissertation

Efficient and Trustworthy Algorithm Design for Large-Scale Simulation and Inference

By

Jingxu Xu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Zeyu Zheng, Chair
Professor Rhonda L. Righter
Assistant Professor Luyi Yang

Spring 2024

Efficient and Trustworthy Algorithm Design for Large-Scale Simulation and Inference

Abstract

Efficient and Trustworthy Algorithm Design for Large-Scale Simulation and Inference

by

Jingxu Xu

Doctor of Philosophy in Engineering - Industrial Engineering and Operations Research

University of California, Berkeley

Assistant Professor Zeyu Zheng, Chair

In the areas of operations research and industrial engineering, the objective function in many optimization and decision-making problems involves complicated system measures that need to be evaluated by simulation. The procedures of evaluating and optimizing stochastic systems via simulation present several critical challenges: i) The stochastic systems typically involve uncertainties that are captured by a set of input distributions to be estimated from real-world input data, and the estimation error from the input distributions may propagate and cause uncertainties in the evaluation of the objective function; ii) Under complicated system logic, the exact simulation of the objective function can be computationally expensive or impossible; iii) Simulations need to be conducted on multiple values of the decision variable or across various systems to facilitate optimization and selection. These challenges exacerbate issues about the allocation and management of computational resources. This thesis aims to propose effective algorithm design for simulation-based decision making, with the goal of optimizing the stochastic systems and saving computational resources at the same time. In particular, our contributions are threefold: i) With the objective of selecting the system with the best performance, we propose a general framework to analyze the joint resource allocation problem for collecting input data and generating simulation replications; ii) For continuous optimization via simulation (COvS) problems, we propose gradient-based algorithms that sequentially utilizes multi-resolution approximations to optimize systems for which exact simulation is costly or impossible. iii) We provide new time-parallel simulation algorithms, so that simulation can be executed quickly to inform time-sensitive decisions.

Contents

List of Figures

List of Tables

Acknowledgments

I am profoundly grateful for the support and guidance I have received during my PhD journey. First and foremost, I extend my deepest appreciation to my advisor, Prof. Zeyu Zheng, whose expertise and insightful guidance were invaluable to my research and academic growth. He is patient and willing to help with every perspective of research, from research directions to making high-quality slides. Prof. Zheng has not only been a remarkable mentor but also a constant source of encouragement and inspiration. I am truly grateful for his mentorship and unwavering support.

I am also thankful for the unwavering support and constructive critiques and feedback from the members of my dissertation committee, Prof. Rhonda L. Righter and Prof. Luyi Yang. Their evaluations, suggestions and guidance have significantly helped me address the deficiencies in my research projects.

I would also like to acknowledge Prof. Anil Aswani, for serving on my qualifying exam committee. His guidance and critiques have helped me improve my work, which was finally published in a journal. I am thankful for Prof. Rhonda L. Righter and Prof. Thibaut Mastrolia, since I am very fortunate to be teaching assistant in their lessons. I appreciate their great efforts on teaching, and learn a lot from them about how to be a good teacher.

I owe a great deal of gratitude to my seniors, peers and friends in Berkeley: Yunduan Lin, Heyuan Liu, Yuhao Ding, Hansheng Jiang, Jiaming Wang, Haoting Zhang, Hyungki Im, Meng Li, Tor Nitayanont, Yuhang Wu, Ziyang Liu, Donghao Ying, Mengzi Guo, Yunkai Zhang, Shuo Sun, Runhan Xie, Jinghai He, Hao Wang. They have provided both moral support and much-needed respite from the demands of my studies. Their encouragement and companionship have been a constant source of strength. Particularly I want to acknowledge my previous roommates, Haoting Zhang and Jiaming Wang. They helped me a lot during the difficult period of COVID. Living with them has brought laughter and relief during the most stressful times.

Last but certainly not least, I must express my heartfelt thanks to my family, especially my parents. Their endless love and encouragement have been the cornerstone of my achievements. I am eternally grateful for their sacrifices and unwavering belief in my potential.

Chapter 1

Introduction

In the areas of industrial engineering and operations research, the need to compare multiple stochastic systems, or optimize a single stochastic system, is a common challenge due to the complexity of such systems. Simulation serves as an important and commonly employed tool to evaluate the performance of complex stochastic systems and inform decisions. For example, in the fields of transportation and logistics, simulation is applied for traffic signal timing optimization [111, 14]. In the fields of manufacturing, simulation is applied to optimally specifying inventory management policies and stock levels for supply chain networks [88]. In the fields of healthcare, simulation is applied to comparing the effectiveness of different configurations of servers and facilities in hospitals [19]. In this thesis, we focus on two decision-making tasks based on simulation: i) selecting the system with the best expected performance, ii) optimizing the expected value of a stochastic objective function with continuous multi-dimensional decision variable. The two tasks are often referred to within the framework of *simulation optimization*: see [10, 80, 28, 59, 106, 115, 63, 57, 15, 53].

In the context of simulation-based decision making, the management and allocation of computing resources consistently emerge as an essential consideration. Several pertinent concerns are outlined below. Firstly, the stochastic systems typically involve uncertainties that are captured by a set of probability models. The associated probability distributions are estimated from data, and there exists a statistical estimation error due to the finite amount of data collected. The statistical estimation error from the input distributions may propagate and cause uncertainties in the evaluation of system performance, known as *input uncertainty*. Given that both data collection and simulation are costly, optimal resource allocation strikes a balance between input data collection and simulation. Secondly, for complicated stochastic systems, the exact simulation of system performance can be computationally expensive or even impossible. For associated optimization of the system, the computational resource need to be allocated to implement the difficult simulation on multiple values of the decision variable to find a good solution. Thirdly, simulation can be time-sensitive, specially when we want to evaluate the performance of the system over a long time horizon. Contrarily, many decisions need to be made in real time and are time-sensitive. This poses requirements on parallel simulation with the challenge of efficiently managing multiple processors.

This thesis targets to make advancements in addressing these concerns and provide new algorithm designs that achieve optimal balance between accuracy and resource efficiency. In the following sections of this chapter, we present a brief summary of our contributions, as well as related publications and working paper referenced throughout the dissertation.

1.1 Summary of Contributions

In this section, we provide a brief summary of our contributions to the simulation literature.

- **Chapter 2**

  With the objective of selecting the system with the best performance, we propose a general framework to analyze the joint resource allocation problem for collecting input data and generating simulation replications. Two commonly arised features, correlation in input data and common random numbers in simulation, are jointly exploited to save costs. For input data collection, in presence of the correlation structure among different sources, options are available to either jointly collect data simultaneously from the different sources, or to collect data solely from a particular source. For simulation, one has the option to either use common random numbers to simultaneously evaluate performance for different systems or to evaluate independently a single system. We provide closed-form optimal resource allocation solutions that maximize the asymptotic probability of correct selection, given fixed resource budget. Our results explicitly show that how the correlation structure is exploited to save costs and improve performance, and how the optimal resource allocation strategy depends on the correlation structure.

- **Chapter 3**

  We propose gradient-based simulation-optimization algorithms to optimize systems that have complicated stochastic structure. The presence of complicated stochastic structure, such as the involvement of infinite-dimensional continuous-time stochastic processes, may cause the exact simulation of the system to be costly or even impossible. On the other hand, for a complicated system, one can sometimes construct a sequence of approximations at different resolutions, where the sequence has finer and finer approximation resolution but higher and higher cost to simulate. With the goal of optimizing the complicated system, we propose algorithms that strategically use the approximations with increasing resolution and higher simulation cost to construct stochastic gradients and perform gradient search in the decision space. To accommodate scenarios where approximations cause discontinuities and lead path-wise gradient estimators to have an uncontrollable bias, stochastic gradients for the proposed algorithms are constructed through finite difference. As a theory support, we prove algorithm convergence rate, central limit theorem, and optimality of algorithm design under the assumption that the objective function for the complicated system is strongly convex, while no such assumptions are imposed on the approximations of the complicated system. We then present a multilevel version of the proposed algorithms to further improve convergence rates, when in addition the sequence of approximations can be naturally coupled.

- **Chapter 4**

  Simulations on single processors can be time-consuming and often need to be completed within short time frames. To address this challenge, we explore the design of algorithms for time-parallel simulation of stochastic models, specifically non-homogeneous Markov chains. Our objective is to efficiently estimate the expected performance of these Markov chains over extended periods within constrained simulation times. We introduce a framework that allocates specific simulation tasks to individual processors and aggregates the simulation results. As a theory support, we prove central limit theorem, and optimality of assignment policy under the assumption that the state space of the Markov chain is finite. We also develop a two-stage parallel simulation procedure that dynamically learns and applies the optimal assignment policy during the simulation period. We prove theoretically and then show empirically that the proposed two-stage procedure improves the simulation efficiency compared to standard assignment policy.

1.2 Related Publications and Working Papers

- **Chapter 2**

  Main Paper:

  – Jingxu Xu, Zeyu Zheng, and Peter W. Glynn.“Joint Resource Allocation for Input Data Collection and Simulation”, 2020 Winter Simulation Conference (WSC). IEEE, 2020.

- **Chapter 3**

  Main Paper:

  – Jingxu Xu, and Zeyu Zheng. “Gradient-based Simulation Optimization Algorithms via Multi-Resolution System Approximations”, INFORMS Journal on Computing 35.3 (2023): 633-651.

  Related Paper:

  – Haoting Zhang, Jinghai He, Jingxu Xu, Jingshen Wang, and Zeyu Zheng. “Enhancing Language Model with Both Human and Artificial Intelligence Feedback Data”, submitted to 2024 Winter Simulation Conference (WSC).

- **Chapter 4**

- Jingxu Xu, and Zeyu Zheng. “Optimal Policies for Time-Parallel Simulation of Time-Varing Systems”, working paper.

Chapter 2

Joint Resource Allocation for Data Collection and Simulation

The need to compare the expected performance of two or multiple stochastic systems naturally arise in the areas of healthcare, supply chain, logistics, production, queueing systems, portfolio management, among others. The stochastic systems under consideration typically involve uncertainties that are captured by a set of probability models. The associated probability distributions are either specified by domain experts or estimated from data. These probability models, serving as inputs to the stochastic system, can be the customer arrival and service processes in a service system or the daily demands and lead times in a supply chain system. After the input probability distributions are specified or estimated, simulation is often used to evaluate the expected system performance, particularly in cases where analytical solutions are not available.

In this chapter, we presumes that there is independent and identically distributed data available or that can be collected that faithfully represents the true input distributions. When the input probability distributions are estimated from data, there exists a statistical estimation error due to the finite amount of data collected. The statistical estimation error from the input distributions may propagate and cause uncertainties in the evaluation of system performance. The resulting uncertainty in the performance evaluation caused by using an incorrectly specified input distribution (due to estimation error) is called *input uncertainty.* The input uncertainty cannot be eliminated by increasing the number of simulation replications used to evaluate the system performance, but can potentially be reduced by collecting more data. In many applications, the data to be collected is multi-dimensional and is usually generated sequentially from real operations. Therefore typically the available data that can be collected in one operational period is corrected. Consider a site selection problem where a manager chooses one out of m sites to run a new branch store. The manager may need to run simulation to evaluate revenues for each site with different store designs. One of the key input distributions is the daily traffic flows. On day i, the full set of data that may be collected is $\mathcal{A}_i = (A_{i,1}, A_{i,2}, \ldots, A_{i,m})$ in which $A_{i,j}$ denotes the traffic flow at the j-th site on day i. Due to the nature of the data generation process, $(A_{i,1}, A_{i,2}, \ldots, A_{i,m})$

are typically correlated, possibly due to common unobserved features from the same day and other observed dependencies such as the adjacency between sites. Ideally, collecting this full set of data $\mathcal{A}_i$ is preferred and provides the best possible information, but this can be costly or even inaccessible. On the other hand, when the correlation structure within the full set of input data generated in the same period is exploited, it may not be necessary to collect the full set of data. Therefore this correlation structure may be used to save input data collection cost.

In this chapter, we propose a general framework to study the joint resource allocation problem for input data collection and simulation. The objective is appropriately allocating resource to maximize the probability of correctly selecting the system with the best performance. Two commonly arised features, correlation in input data and common random numbers in simulation, are jointly exploited to save costs. For input data collection, in presence of the correlation structure among different sources, options are available to either jointly collect data simultaneously from the different sources, or to collect data solely from a particular source. For simulation, one has the option to either use common random numbers to simultaneously evaluate performance for different systems or to evaluate independently a single system. We provide closed-form optimal resource allocation solutions that maximize the asymptotic probability of correct selection, given fixed resource budget. Our results explicitly show that how the correlation structure is exploited to save costs and improve performance, and how the optimal resource allocation strategy depends on the correlation structure.

Two scenarios are considered in our framework. First, we consider scenarios (in Section 2.2) where the “monetary” cost of generating a simulation replication is much smaller than the cost of collecting a sample of input data. For example, these situations happen when the scale and structure of the problem permits the generation of simulation replications efficiently even on personal computers, while the input data needs to be purchased from a data vendor at a significant price or needs to be collected by multiple staff throughout a number of days. One may then assume a simplification that once the input distributions are estimated, the expected performance evaluation via simulation is immediately available at no cost. Therefore in these situations, the resource allocation problem focuses entirely on the input data collection part. Second, we consider scenarios (in Section 2.4) where the simulation cost is not negligible compared with the input data collection cost. These scenarios arise in performance evaluation for complicated systems, in which high performance computing resources are needed for simulation. Otherwise if not using designated high performance computing resources, the simulation may take too long a time. The simulation costs therefore may be evaluated by monetary costs for purchasing computing resources or by the opportunity costs for long simulation time. In these scenarios, the optimal resource allocation strikes a balance between the input data collection costs and simulation costs, to jointly control input uncertainty and simulation error.

2.1 Related Work

Comparing the expected performance of two or multiple systems via simulation is a fundamental component in the problems of Ranking and Selection (R&S) and Discrete Optimization via Simulation (DOvS). When one knows explicitly the input distributions, or has the ability to generate simulation replications from the true input distributions, the focus of these problems is then on developing efficient simulation procedures to select the best system. Procedures developed in the literature typically adopt a frequentist or a Bayesian view. See [65] and [11] for an overview. This chapter follows a frequentist perspective.

When the input distributions are not explicitly known, or when one does not have the ability to generate samples from the true input distributions, a series of works discuss the quantification of input uncertainty and its impact on the comparison of system performance. See [12, 13, 5, 104, 91, 16, 102, 101] among others. The closest to our work are [102] and [90]. Song and Nelson [90] exploits the effect of common input distribution to reduce the uncertainty in system performance comparison, when the input data set is given. They construct valid confidence intervals for system comparison that incorporate input uncertainty, the common input distribution effect, and simulation uncertainty. Wu and Zhou [102] allows the collection of additional input data from multiple independent sources and discusses the optimal resource allocation for input data collection and simulation. Our work is different to the literature in three folds: (1) we exploit the correlation structure in the input data to reduce the input data collection cost, and show how the correlation structure impacts the optimal resource allocation strategy. (2) we propose a general framework that integrates input data collection and simulation in which the data collection and simulation costs themselves can be random; (3) we investigate the joint optimal resource allocation when both the correlation in the input data and the use of common random numbers in the simulation procedures are exploited.

When the performance of each system is evaluated independently, the best system selection problem shares the same formulation as the *best arm identification* problem; see [61, 62, 38, 87, 86] among others. In fact, if we separate the input data collection problem and the best system selection problem, each problem shares a very similar formulation with a best arm identification problem, provided that the observation of each dimension in the data is independent and that the performance evaluation of each system is independent. Differences emerge when correlation is present either in the input data collection or among simulation evaluations.

The use of common random numbers (CRN) in the simulation procedures for R& S and DOvS have been widely discussed. See [109, 37, 77, 17, 66] among others. Specifically, Fu et al. [30] discusses the optimal allocation of simulation replications on each system when the CRN technique is used. Their work did not discuss errors created from input data. We propose an alternative framework that allows the simulation costs to be random, and consider the joint resource allocation problem for both input data collection and simulation.

2.2 A General Framework

We introduce a basic and general framework that allows us to integrate simulation for performance evaluation and input data collection for input distribution estimation. In this framework, the costs for simulation and input data collection can be random. We first focus on describing the input data collection and input distribution estimation in the framework.

Consider a set of systems labeled by index set $[m] = \{1, 2, \ldots, m\}$. When $m = 2$, for example, there are two systems to compare. For $i \in [m]$, the expected performance for system i is given by $\alpha_i(\theta_i^*)$, where $\alpha_i : \mathbb{R}^{d_i} \to \mathbb{R}$ is a continuously differentiable function, and $\theta_i^* \in \mathbb{R}^{d_i}$ is the true input distribution parameter associated with system i (e.g., arrival and service rates, lead time expectation, shape parameters, etc.). We first consider scenarios where the simulation cost is negligible compared to input data collection cost, so that the expected performance function $\alpha_i(\cdot)$'s are viewed to be available at no cost whenever the input distribution is specified. We recognize that the input distribution parameters $\theta_1^*, \theta_2^*, \ldots, \theta_m^*$ need to be estimated. This can be based on common observations $(\tilde{X}_{ij} : i \in [m], j \geq 1)$ or based on independently gathered observations $(X_{ij} : j \geq 1)$ for $i \in [m]$. We assume that the cost of collecting the j-th copy of a set of common observations $(\tilde{X}_{ij} : i \in [m])$ collected simultaneously is given by $\tilde{\tau}_j$, while the cost of collecting individual X_{ij} is given by τ_{ij}. These data collection costs can be random variables themselves. Then, with a budget c in hand, we can either collect

$$\tilde{N}(c) = \max\{n \geq 0 : \tilde{\tau}_1 + \ldots + \tilde{\tau}_n \leq c\}$$

copies of "common observations" or

$$N_i(c) = \max\{n \geq 0 : \tau_{i1} + \ldots + \tau_{in} \leq c\}$$

copies of observations solely from system i. In this framework, we assume that

1. $(\tilde{\tau}_j, (\tilde{X}_{ij} : i \in [m]) : j \geq 1)$, $(\tau_{1j}, X_{1j} : j \geq 1)$, $\ldots$, $(\tau_{mj}, X_{mj} : j \geq 1)$ are independent sequences.

2. $(\tau_i, (\tilde{X}_{ij} : i \in [m]) : j \geq 1)$ is i.i.d. in j.

3. For each $i \in [m]$, $((\tau_{ij}, X_{ij}) : j \geq 1)$ is i.i.d. in j.

4. $\tilde{X}_{i1} \overset{\mathcal{D}}{=} X_{i1}$ for $i \in [m]$. $\mathrm{Var}(X_{i1}) < \infty$ for $i \in [m]$.

5. $\mathbb{E}\,\tilde{\tau}_1 < \infty$, $\mathbb{E}\,\tau_{i1} < \infty$, for $i \in [m]$.

In practice, it is often the case that $\mathbb{E}\,\tau_{i1} < \mathbb{E}\,\tilde{\tau}_1$ for $i \in [m]$ and $\mathbb{E}\,\tilde{\tau}_1 \leq \sum_{i=1}^{m} \mathbb{E}\,\tau_{i1}$. Since $\tilde{N}(\cdot)$ and $N_i(\cdot)$'s are renewal counting processes, it is known that

$$\frac{1}{c}\tilde{N}(c) \overset{a.s.}{\to} \lambda \triangleq \frac{1}{\mathbb{E}\,\tilde{\tau}_1}$$

and

$$\frac{1}{c}N_i(c) \overset{a.s.}{\to} \lambda_i \triangleq \frac{1}{\mathbb{E}\,\tau_{i1}}.$$

Given an overall budget c, suppose we allocate a fraction p to collecting common observations, and a fraction p_i to collecting independent observations from system i, where

$$p + p_1 + p_2 + \ldots + p_m = 1,$$

with $p \geq 0, p_i \geq 0, i \in [m]$.

We now assume that one estimates θ_i^* via an maximum likelihood estimator (MLE) $\hat{\theta}_i$, where $\hat{\theta}_i$ maximizes the likelihood

$$\prod_{j=1}^{\tilde{N}(pc)} f_i(\theta, \tilde{X}_{ij}) \prod_{j=1}^{N_i(p_i c)} f_i(\theta, X_{ij}),$$

where $f_i(\theta, \cdot)$ is the marginal probability density function of the input data for the i'th system. By taking the logarithm of the likelihood function

$$\tilde{L}_{ij}(\theta) = \log f_i(\theta, \tilde{X}_{ij}),$$
$$L_{ij}(\theta) = \log f_i(\theta, X_{ij}),$$

maximizing the likelihood is equivalent to maximizing the log-likelihood, given by

$$\sum_{j=1}^{\tilde{N}(pc)} \tilde{L}_{ij}(\theta) + \sum_{j=1}^{N_i(p_i c)} L_{ij}(\theta).$$

Under appropriate technical conditions, the maximum likelihood estimator $\hat{\theta}_i$ satisfies

$$\sum_{j=1}^{\tilde{N}(pc)} \nabla \tilde{L}_{ij}(\hat{\theta}_i) + \sum_{j=1}^{N_i(p_i c)} \nabla L_{ij}(\hat{\theta}_i) = 0$$

with $\hat{\theta}_i \to \theta_i^*$ almost surely as $c \to \infty$. Note that

$$\sum_{j=1}^{\tilde{N}(pc)} \left[\nabla \tilde{L}_{ij}(\hat{\theta}_i) - \nabla \tilde{L}_{ij}(\theta_i^*)\right] + \sum_{j=1}^{N_i(p_i c)} \left[\nabla L_{ij}(\hat{\theta}_i) - \nabla L_{ij}(\theta_i^*)\right] = -\sum_{j=1}^{\tilde{N}(pc)} \nabla \tilde{L}_{ij}(\theta_i^*) - \sum_{j=1}^{N_i(p_i c)} \nabla L_{ij}(\theta_i^*).$$

We adopt the convention that the gradient is a row vector. If $L_{ij}(\cdot)$ is appropriately smooth, then the mean value theorem implies that

$$\sqrt{c}(\hat{\theta}_i - \theta_i^*)\Big(\frac{\tilde{N}(pc)}{c}H_i + \frac{N_i(p_i c)}{c}H_i + o_p(1)\Big) = -\frac{\sum_{j=1}^{\tilde{N}(pc)} \nabla \tilde{L}_{ij}(\theta_i^*)}{\sqrt{c}} - \frac{\sum_{j=1}^{N_i(p_i c)} \nabla L_{ij}(\theta_i^*)}{\sqrt{c}},$$

where the notion $o_P(1)$ indicates a small random quantity that weakly converges to zero as $c \to \infty$. The Hessian matrix H_i is given by

$$H_i = \big(\mathbb{E}\, \frac{\partial^2}{\partial\theta_k \partial\theta_l} L_{i1}(\theta^*) : 1 \le k \le l \le d_i\big).$$

Assume that H_i is negative definite so that $\det H_i$ is non-singular. Then, when c is large,

$$c^{\frac{1}{2}}(\hat{\theta}_i - \theta_i^*) = -\frac{1}{\lambda p + \lambda_i p_i}\Big(\sum_{j=1}^{\tilde{N}(pc)} \frac{\nabla \tilde{L}_{ij}(\theta^*) H_i^{-1}}{\sqrt{c}} + \sum_{j=1}^{N_i(p_i c)} \frac{\nabla L_{ij}(\theta^*) H_i^{-1}}{\sqrt{c}}\Big) + o_p(1).$$

Hence, we have the following result.

Theorem 1. *Assume that for $i \in [m]$, there exists an open subset w_i that is a subset of the feasible parameter region where the true parameter $\theta_i^* \in w_i$, and that all third-order partial derivatives of the log-likelihood functions with respect to input parameters are uniformly bounded for $\theta_i \in w_i$. When $c \to \infty$,*

$$c^{\frac{1}{2}}((\hat{\theta}_i - \theta_i^*) : i \in [m]) \Rightarrow \Big(-\frac{1}{\lambda p + \lambda_i p_i}\big(\sqrt{\lambda p}\, \tilde{G}_i + \sqrt{\lambda_i p_i}\, G_i\big) : i \in [m]\Big),$$

where:

- *$(\tilde{G}_1, \ldots, \tilde{G}_m)$ is jointly Gaussian with mean 0.*
- *$\tilde{G}_i \overset{\mathcal{D}}{=} G_i$, where the covariance matrix of G_i is given by $H_i^{-1}\, \mathbb{E}\, \nabla L_{i1}(\theta^*)^\top \nabla L_{i1}(\theta^*) H_i^{-1}$.*
- *The random variables (rv's) $G_1, G_2, \ldots, G_m$ are independent and independent of $(\tilde{G}_1, \ldots, \tilde{G}_m)$.*

The proof of Theorem 1 is given in the appendices. With Theorem 1 in hand,

$$\begin{aligned} c^{\frac{1}{2}}(\alpha_i(\hat{\theta}_i) - \alpha_i(\theta_i^*)) &= (\hat{\theta}_i - \theta_i^*)\nabla\alpha_i(\theta^*)^\top + o_p(1) \\ &= -\frac{1}{\lambda_p + \lambda_i p_i}(\sqrt{\lambda p}\tilde{G}_i + \sqrt{\lambda_i p_i} G_i)\nabla\alpha_i(\theta^*)^\top + o_p(1) \\ &= -\frac{1}{\lambda_p + \lambda_i p_i}(\sqrt{\lambda p}\tilde{\mathcal{G}}_i + \sqrt{\lambda_i p_i}\mathcal{G}_i) + o_p(1) \end{aligned}$$

as $c \to \infty$, where $\tilde{\mathcal{G}}_i = \tilde{G}_i \nabla\alpha_i(\theta^*)^\top, \mathcal{G}_i = G_i \nabla\alpha_i(\theta^*)^\top, i \in [m]$. Note that $\tilde{\mathcal{G}}_i \overset{\mathcal{D}}{=} \mathcal{G}_i, i \in [m]$. Hence, when $c \to \infty$,

$$c^{\frac{1}{2}}(\alpha_i(\hat{\theta}_i) - \alpha_j(\hat{\theta}_j) - (\alpha_i(\theta_i^*) - \alpha_j(\theta_j^*)) \Rightarrow -\frac{\sqrt{\lambda p}}{\lambda p + \lambda_i p_i}\tilde{\mathcal{G}}_i + \frac{\sqrt{\lambda p}}{\lambda p + \lambda_j p_j}\tilde{\mathcal{G}}_j - \frac{\sqrt{\lambda_i p_i}}{\lambda p + \lambda_i p_i}\mathcal{G}_i + \frac{\sqrt{\lambda_j p_j}}{\lambda p + \lambda_j p_j}\mathcal{G}_i.$$

Denote the right-hand-side above as W_{ij}. The random variable (rv) W_{ij} is Gaussian with mean zero and variance

$$\frac{\sigma_i^2}{\lambda p + \lambda_i p_i} + \frac{\sigma_j^2}{\lambda p + \lambda_j p_j} - \frac{2\lambda p\, c_{ij}}{(\lambda p + \lambda_i p_i)(\lambda p + \lambda_j p_j)},$$

where $\sigma_i^2 = \operatorname{Var} \mathcal{G}_i$ and $c_{ij} = \operatorname{Cov}(\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}_j)$. We further define $\rho_{ij} = \operatorname{Corr}(\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}_j)$.

With the given framework, when comparing systems i and j, the input data collection problem can be summarized into the following optimization problem

$$\min_{p\geq 0,\, p_i \geq 0,\, p_j \geq 0,\, p+p_i+p_j=1} \frac{\sigma_i^2}{\lambda p + \lambda_i p_i} + \frac{\sigma_j^2}{\lambda p + \lambda_j p_j} - \frac{2\lambda p\, c_{ij}}{(\lambda p + \lambda_i p_i)(\lambda p + \lambda_j p_j)}.$$

Recall that $c_{ij} = \rho_{ij}\sigma_i\sigma_j$. An equivalent formulation is to set $q_i = \lambda_i p_i$, $q = \lambda p$, and minimize

$$\frac{\sigma_i^2}{q + q_i} + \frac{\sigma_j^2}{q + q_j} - \frac{2q\rho_{ij}\sigma_i\sigma_j}{(q+q_i)(q+q_j)} \tag{2.1}$$

subject to $\frac{q}{\lambda} + \frac{q_i}{\lambda_i} + \frac{q_j}{\lambda_j} = 1$.

2.3 Optimal Resource Allocation for Input Data Collection

The optimization problem given by (2.1) to solve the optimal resource allocation for input data collection turns out to be non-convex. The non-convexity is exactly caused by the correlation feature and creates difficulty in modeling how the correlation ρ_{ij} presented in the input data exactly affects the optimal resource allocation. The following theorem shows that the optimal objective function can be obtained at the boundary of feasible region $\mathcal{S} = \{(q_i, q_j, q)^\top : q_i, q_j, q \geq 0, \frac{q}{\lambda} + \frac{q_i}{\lambda_i} + \frac{q_j}{\lambda_j} = 1\} \subset \mathbb{R}^3$.

Theorem 2. *For the optimization problem* (2.1)*, there exists a solution* (q_i^*, q_j^*, q^*) *that achieves the optimal objective value and has at least one element as zero.*

With Theorem 2 in hand, it suffices to explore the resource allocation strategies among $(q_i, 0, q)$, $(q_i, q_j, 0)$ and $(0, q_j, q)$. Denote $s_i := \frac{1}{\lambda_i}$, $s_j := \frac{1}{\lambda_j}$, $s := \frac{1}{\lambda}$, $v(q_i, q_j, q) := \frac{\sigma_i^2}{q_i} + \frac{\sigma_j^2}{q_j} - \frac{2q\rho_{ij}\sigma_i\sigma_j}{(q+q_i)(q+q_j)}$. Theorem 2 implies that

$$\min_{(q_i,q_j,q)^\top \in \mathcal{S}} v(q_i, q_j, q) = \min\left\{b_i^*, b_j^*, \tilde{b}^*\right\},$$

where $b_i^* = \min\limits_{(q_i,q_j,q)^\top \in \mathcal{S}, q_i=0} v(q_i, q_j, q)$, $b_j^* = \min\limits_{(q_i,q_j,q)^\top \in \mathcal{S}, q_j=0} v(q_i, q_j, q)$, $\tilde{b}^* = \min\limits_{(q_i,q_j,q)^\top \in \mathcal{S}, q=0} v(q_i, q_j, q)$. We provide the closed-form value for $b_i^*, b_j^*, \tilde{b}^*$ and the associated optimizers.

$$b_i^* = \begin{cases} \left(\sqrt{(-s_j + s)\,\sigma_i^2} + \sqrt{s_j(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j)}\right)^2, \text{ if } s_j\sigma_i^2 < \left(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j\right)(s - s_j), \\ s(\sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j), \text{ if } s_j\sigma_i^2 \geq \left(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j\right)(s - s_j). \end{cases}$$

The optimal solution to achieve b_i^* is

$$
\begin{aligned}
(q_i, q_j, q) = \Bigg(0,\ & \frac{\sqrt{(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_j)} - \sqrt{s_j\sigma_i^2}}{(s - s_j)\sqrt{s_j\sigma_i^2} + s_j\sqrt{(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j)(s - s_j)}}, \\
& \frac{1}{s} - \frac{s_j\sqrt{\left(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j\right)(s - s_j)} - \sqrt{s_j^3\sigma_i^2}}{s\,(s - s_j)\,\sqrt{s_j\sigma_i^2} + s_j s\sqrt{(\sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j)(s - s_j)}} \Bigg)
\end{aligned}
$$

and $(q_i, q_j, q) = (0, 0, \frac{1}{s})$ respectively under the two conditions.

$$
b_j^* = \begin{cases} \left(\sqrt{(-s_i + s)\,\sigma_j^2} + \sqrt{s_i(\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)}\right)^2, & \text{if } s_i\sigma_j^2 < (\sigma_i^2 - 2\rho_{ij}\sigma\sigma_j)\,(s - s_i)\,, \\ s(\sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j), & \text{if } s_i\sigma_j^2 \geq (\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_i)\,. \end{cases}
$$

The optimal solution to achieve b_j^* is

$$
\begin{aligned}
(q_i, q_j, q) = \Bigg(& \frac{\sqrt{(\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_i)} - \sqrt{s_i\sigma_j^2}}{(s - s_i)\,\sqrt{s_i\sigma_j^2} + s_i\sqrt{(\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_i)}}, 0, \\
& \frac{1}{s} - \frac{s_i\sqrt{(\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_i)} - \sqrt{s_i^3\sigma_j^2}}{s\,(s - s_i)\,\sqrt{s_i\sigma_j^2} + s_i s\sqrt{(\sigma_i^2 - 2\rho_{ij}\sigma_i\sigma_j)\,(s - s_i)}} \Bigg)
\end{aligned}
$$

and $(q_i, q_j, q) = (0, 0, \frac{1}{s})$ respectively under the two conditions.

$$
\tilde{b}^* = (\sqrt{s_i}\sigma_i + \sqrt{s_j}\sigma_j)^2.
$$

The optimal solution to achieve $\tilde{b}^*$ is $(q_i, q_j, q) = (\frac{\sigma_i}{\sqrt{s_i}(\sqrt{s_i}\sigma_i + \sqrt{s_j}\sigma_j)}, \frac{\sigma_j}{\sqrt{s_j}(\sqrt{s_i}\sigma_i + \sqrt{s_j}\sigma_j)}, 0)$.

As shown in the closed-form solution for $\min\limits_{(q_i,q_j,q)^\top \in \mathcal{S}} v(q_i, q_j, q)$, the optimal resource allocation critically depends on the sign and magnitude of the correlation ρ_{ij}. We discuss how the different value of ρ_{ij} affects the optimal allocation. First, consider the case when $s = s_i + s_j$. This corresponds to the case of additive input data collection costs. If $\rho_{ij} \leq 0$, the optimal allocation is to independently collect data for system i and j, and the optimal fraction q_i^*, q_j^* satisfies $\frac{\sigma_i}{q_i^*\sqrt{s_i}} = \frac{\sigma_j}{q_j^*\sqrt{s_j}}$. Specifically if $\rho_{ij} = 0$, any allocation (q, q_i, q_j) that satisfies $\frac{\sigma_i}{\sqrt{s_i}(q_i+q)} = \frac{\sigma_j}{\sqrt{s_j}(q_j+q)}$ is optimal. If $0 < \rho_{ij} < \max\left\{\frac{1}{2\sigma_i\sigma_j}(\sigma_j^2 - \frac{s_j\sigma_i^2}{s_i}), \frac{1}{2\sigma_i\sigma_j}(\sigma_i^2 - \frac{s_i\sigma_j^2}{s_j})\right\}$, the optimal allocation assigns a fraction of budget to collecting common observations and assigns the rest of budget to solely collecting data from system i if $\sigma_i^2/s_i > \sigma_j^2/s_j$, or solely system j if otherwise $\sigma_j^2/s_j > \sigma_i^2/s_i$. When $\rho_{ij} \geq \max\left\{\frac{1}{2\sigma_i\sigma_j}(\sigma_i^2 - \frac{s_i\sigma_j^2}{s_j}), \frac{1}{2\sigma_i\sigma_j}(\sigma_j^2 - \frac{s_j\sigma_i^2}{s_i})\right\}$, the optimal allocation assigns all the budget to collecting common observations of data. In summary, when $s = s_i + s_j$, there are three different regimes of optimal allocation depending on the sign and value of ρ_{ij}.

When $s < s_i + s_j$, two critical values that affect the optimal allocation are $s/\min\{s_i, s_j\}$ and ρ_{ij}. Note that in practice $s/\min\{s_i, s_j\}$ is always greater than 1. Theorem 2 shows that there exists a threshold value $\gamma > 1$ such that if $s/\min\{s_i, s_j\} > \gamma$, the optimal allocation strategy adopts three different regimes. When ρ_{12} is close to -1, the optimal allocation assigns all the budget to independently collecting data for system i and j. As ρ_{ij} increases, the optimal allocation assigns a fraction of budget to collecting common observations of data simultaneously and assigns the rest budget to independently collecting data from one of the two systems (in a way analogous to the case of $s = s_i + s_j$). When ρ_{ij} further increases and exceeds $\max\left\{\frac{1}{2\sigma_i\sigma_j}(\sigma_j^2 - \frac{s_j\sigma_i^2}{s-s_j}), \frac{1}{2\sigma_i\sigma_j}(\sigma_i^2 - \frac{s_i\sigma_j^2}{s-s_i})\right\}$, the optimal allocation is to collecting common observations of data simultaneously. The above summarizes the three regimes for scenarios where $s/\min\{s_i, s_j\} > \gamma$. On the other hand, if $1 < s/\min\{s_i, s_j\} < \gamma$, there are two different forms. Specifically, there exists a threshold $\rho' < 0$ such that, if $-1 < \rho_{ij} < \rho'$, the optimal allocation assigns the budget to collecting data independently for system i and j; if $\rho_{ij} > \rho'$, the optimal allocation assigns all the budget to collecting common observations of data.

2.4 Joint Resource Allocation Formula

In this section, we consider scenarios where the simulation cost is not negligible compared with the input data collection cost. These scenarios arise in performance evaluation for complicated systems, in which high performance computing resources are needed. The simulation may take too long time if not using designated high performance computing resources. The simulation costs therefore may be evaluated by monetary costs for purchasing computing resources or by the opportunity costs for long simulation time. In these scenarios, it is unrealistic to assume that the expected performance $\alpha_i(\cdot)$'s are immediately available at negligible cost. In this section, we extend the general framework introduced in Section 3 to include both input data collection and simulation generation.

Recall that as defined in Section 3, the expected performance of system i is given by $\alpha_i(\theta_i^*)$ for $i \in [m]$, where $\theta_i^* \in \mathbb{R}^{d_i}$ is the true input distribution parameter. Input data is collected and used to derive maximum likelihood estimators for θ_i^*'s, denoted by $\hat{\theta}_i$ for $i \in [m]$. Simulation needs to be run to estimate the expected performance $\alpha_i(\hat{\theta}_i)$'s given the estimated input distribution parameters. The simulation can be done by independently running system i and the sequence of simulation output is $(Y_{ij}(\hat{\theta}_i) : j \geq 1)$ for $i \in [m]$. Alternatively, the technique of common random numbers (CRN) can be used to evaluate the m systems simultaneously. The sequence of simulation output using CRN is $((\tilde{Y}_{ij}(\hat{\theta}) : i \in [m]) : j \geq 1)$. When using CRN, we assume that the cost of obtaining the j-th simulation replication of a set of simultaneous evaluations $(\tilde{Y}_{1j}(\hat{\theta}), \tilde{Y}_{2j}(\hat{\theta}), ...\tilde{Y}_{mj}(\hat{\theta}))$ is given by $\tilde{\eta}_j$, and the cost of generating individual evaluation $Y_{ij}(\hat{\theta}_i)$ for system i is given by η_{ij}. The simulation costs can be random. Then, given a simulation cost budget $\tilde{c}$, we can either generate

$$\tilde{M}(\tilde{c}) = \max\{n \geq 0 : \tilde{\eta}_1 + \ldots + \tilde{\eta}_n \leq \tilde{c}\}$$

simulation replications using CRN or

$$M_i(\tilde{c}) = \max\{n \geq 0 : \eta_{i1} + \ldots + \eta_{in} \leq \tilde{c}\}$$

individual simulation replications for system i. We assume that

1. Conditional on the estimated input distribution parameters $\hat{\theta} = \{\hat{\theta}_i : i \in [m]\}$, $(\tilde{\eta}_j, (\tilde{Y}_{ij}(\hat{\theta}) : i \in [m]) : j \geq 1), (\eta_{1j}, Y_{1j}(\hat{\theta}_1) : j \geq 1), \ldots, (\eta_{mj}, Y_{mj}(\hat{\theta}_m) : j \geq 1)$ are independent sequences.

2. Conditional on $\hat{\theta}$, $(\tilde{\eta}_j, (\tilde{Y}_{ij}(\hat{\theta}) : i \in [m]))$ is independent and identically distributed (iid) in j, and $(\eta_{ij}, Y_{ij}(\hat{\theta}_i))$ is iid in j for each $i \in [m]$.

3. $\mathbb{E}[Y_{i1}|\hat{\theta}_i] = \mathbb{E}[\tilde{Y}_{i1}|\hat{\theta}] = \alpha_i(\hat{\theta}_i)$, and $\mathrm{Var}(Y_{i1}|\hat{\theta}_i) = \mathrm{Var}(\tilde{Y}_{i1}|\hat{\theta}) = D_i(\hat{\theta}_i)$ for $i \in [m]$. $\mathrm{Cov}(\tilde{Y}_{i1}, \tilde{Y}_{j1}) = D_{ij}(\hat{\theta})$ for $i, j \in [m]$, where $D_i(\cdot)$ and $D_{ij}(\cdot)$ are continuous functions with respect to θ_i and θ.

4. $\mathbb{E}\,\tilde{\eta}_1 < \infty$, $\mathbb{E}\,\eta_{i1} < \infty$, for $i \in [m]$. In general, $\mathbb{E}\,\eta_{i1} < \mathbb{E}\,\tilde{\eta}_1$ for $i \in [m]$.

Since $\tilde{M}(\cdot)$ and $M_i(\cdot)$'s are renewal counting processes, we have as $\tilde{c} \to \infty$, $\frac{1}{\tilde{c}}\tilde{M}(\tilde{c}) \overset{a.s.}{\to} \mu \triangleq \frac{1}{\mathbb{E}\,\tilde{\eta}_1}$ and $\frac{1}{\tilde{c}}M_i(\tilde{c}) \overset{a.s.}{\to} \mu_i \triangleq \frac{1}{\mathbb{E}\,\eta_{i1}}$. Given a simulation budget $\tilde{c}$, suppose we allocate a fraction r to simultaneous evaluations using CRN, and a fraction r_i to independent simulation evaluation for system i, where

$$r + r_1 + r_2 + \ldots + r_m = 1,$$

with $r \geq 0$, $r_i \geq 0$, $i \in [m]$.

With a given simulation budget and allocation, the simulation estimators for $\alpha_i(\hat{\theta}_i)$'s are given by

$$\hat{\alpha}_i(\hat{\theta}_i) = \frac{\sum_{j=1}^{\tilde{M}(r\tilde{c})} \tilde{Y}_{ij}(\hat{\theta}) + \sum_{j=1}^{M_i(r_i\tilde{c})} Y_{ij}(\hat{\theta}_i)}{\tilde{M}(r\tilde{c}) + M_i(r_i\tilde{c})}.$$

Then,

$$\tilde{c}^{1/2}(\hat{\alpha}_i(\hat{\theta}_i) - \alpha_i(\hat{\theta}_i)) = -\frac{1}{\mu r + \mu_i r_i}\left(\sqrt{\mu r}\tilde{Z}_i(\hat{\theta}) + \sqrt{\mu_i r_i} Z_i(\hat{\theta}_i)\right) + o_p(1),$$

where, conditional on $\hat{\theta}$,

- $(\tilde{Z}_1(\hat{\theta}), \ldots, \tilde{Z}_m(\hat{\theta}))$ is jointly Gaussian with mean 0. $\mathrm{Var}(\tilde{Z}_i(\hat{\theta})) = D_i(\hat{\theta}_i)$ and $\mathrm{Cov}(\tilde{Z}_i(\hat{\theta}), \tilde{Z}_j(\hat{\theta})) = D_{ij}(\hat{\theta})$ for $1 \leq i \leq j \leq m$.

- $\tilde{Z}_i(\hat{\theta}) \overset{\mathcal{D}}{=} Z_i(\hat{\theta}_i)$, and specifically $\mathrm{Var}(\tilde{Z}_i(\hat{\theta})) = \mathrm{Var}(Z_i(\hat{\theta}_i))$.
- The rv's $Z_1(\hat{\theta}_1), Z_2(\hat{\theta}_2), \ldots, Z_m(\hat{\theta}_m)$ are independent and independent of $(\tilde{Z}_1(\hat{\theta}), \tilde{Z}_2(\hat{\theta}), \ldots, \tilde{Z}_m(\hat{\theta}))$.

Hence, as $\tilde{c} \to \infty$,

$$\begin{aligned}&\tilde{c}^{\frac{1}{2}}(\hat{\alpha}_i(\hat{\theta}_i) - \hat{\alpha}_j(\hat{\theta}_j) - (\alpha_i(\hat{\theta}_i) - \alpha_j(\hat{\theta}_j)))\\ \Rightarrow\ & -\frac{\sqrt{\mu r}}{\mu r + \mu_i r_i}\tilde{Z}_i(\hat{\theta}) + \frac{\sqrt{\mu r}}{\mu r + \mu_j r_j}\tilde{Z}_j(\hat{\theta}) - \frac{\sqrt{\mu_i r_i}}{\mu r + \mu_i r_i}Z_i(\hat{\theta}_i) + \frac{\sqrt{\mu_j r_j}}{\mu r + \mu_j r_j}Z_j(\hat{\theta}_j).\end{aligned}$$

Denote the limiting rv as $V_{ij}(\hat{\theta})$. Conditional on $\hat{\theta}$, the rv $V_{ij}(\hat{\theta})$ is Gaussian with mean zero and variance

$$\frac{D_i(\hat{\theta}_i)}{\mu r + \mu_i r_i} + \frac{D_j(\hat{\theta}_j)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\hat{\theta})}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)}.$$

Therefore, conditional on the input distribution specified by $\hat{\theta}$ and given a simulation budget c, the optimal simulation budget allocation problem in order to differentiate system i and j is given by

$$\min_{r_i \geq 0,\, r_j \geq 0,\, r \geq 0,\, r_i + r_j + r = 1} \quad \frac{D_i(\hat{\theta}_i)}{\mu r + \mu_i r_i} + \frac{D_j(\hat{\theta}_j)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\hat{\theta})}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)}.$$

Suppose that we want to compare the system performance for system i and system j and a total budget C is allocated to both input data collection and simulation experiments. Suppose we allocate a fraction p to collecting common observations, a fraction p_i (or p_j) to collecting independent observations from system i (or j), a fraction r to running simulation replications to evaluate m systems simultaneously using CRN, and a fraction r_i (or r_j) to running individual simulation replications for system i (or j) , where

$$p + p_i + p_j + r + r_i + r_j = 1$$

with $p \geq 0$, $p_i \geq 0$, $r \geq 0$, $r_i \geq 0$, $i \in [m]$. The resulted system performance estimations are $\hat{\alpha}_i(\hat{\theta}_i)$ and $\hat{\alpha}_j(\hat{\theta}_j)$. The following central limit theorem is a direct result by noticing that the uncertainty presented in the input data collection and the uncertainty emerged from simulation replications are independent.

Theorem 3. *When $C \to \infty$,*

$$\begin{aligned}&C^{\frac{1}{2}}[(\hat{\alpha}_i(\hat{\theta}_i) - \hat{\alpha}_j(\hat{\theta}_j)) - (\alpha_i(\theta_i^*) - \alpha_j(\theta_j^*))]\\ =\ & C^{\frac{1}{2}}[(\hat{\alpha}_i(\hat{\theta}_i) - \alpha_i(\hat{\theta}_i)) - (\hat{\alpha}_j(\hat{\theta}_j) - \alpha_j(\hat{\theta}_j))] + C^{\frac{1}{2}}[(\alpha_i(\hat{\theta}_i) - \alpha_i(\theta_i^*)) - (\alpha_j(\hat{\theta}_j) - \alpha_j(\theta_j^*))]\\ \Rightarrow\ & U_{ij}(\theta^*),\end{aligned}$$

where $U_{ij}(\theta^*)$ *is a Gaussian rv with mean zero and variance* $\mathrm{Var}_{ij}(\theta^*)$ *given by*

$$\frac{\sigma_i^2}{\lambda p + \lambda_i p_i} + \frac{\sigma_j^2}{\lambda p + \lambda_j p_j} - \frac{2\lambda p\, c_{ij}}{(\lambda p + \lambda_i p_i)(\lambda p + \lambda_j p_j)} + \\ \frac{D_i(\theta_i^*)}{\mu r + \mu_i r_i} + \frac{D_j(\theta_j^*)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\theta^*)}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)}.$$

When comparing two systems i and j and selecting the better, maximizing the asymptotic probability of correct selection is equivalent to minimizing the limiting variance $\mathrm{Var}_{ij}(\theta^*)$. The associated joint optimal budget allocation problem is given by

$$\begin{aligned} \min_{p_i,p_j,p,r_i,r_j,r} \quad & \mathrm{Var}_{ij}(\theta^*) \\ s.t. \quad & p_i + p_j + p + r_i + r_j + r = 1 \\ & p_i, p_j, p, r_i, r_j, r \geq 0. \end{aligned} \tag{2.2}$$

To solve (2.2), we define $L_1(p_i,p_j,p) := \frac{\sigma_i^2}{\lambda p+\lambda_i p_i} + \frac{\sigma_j^2}{\lambda p+\lambda_j p_j} - \frac{2\lambda p\, c_{ij}}{(\lambda p+\lambda_i p_i)(\lambda p+\lambda_j p_j)}$, $L_2(r_i,r_j,r) := \frac{D_i(\theta_i^*)}{\mu r+\mu_i r_i} + \frac{D_j(\theta_j^*)}{\mu r+\mu_j r_j} - \frac{2\mu r\, D_{ij}(\theta^*)}{(\mu r+\mu_i r_i)(\mu r+\mu_j r_j)}$, and set $w_1 = p_i + p_j + p$, $w_2 = r_i + r_j + r, \tilde{p}_i = \frac{p_i}{w_1}, \tilde{p}_j = \frac{p_j}{w_1}, \tilde{p} = \frac{p}{w_1}, \tilde{r}_i = \frac{r_i}{w_2}, \tilde{r}_j = \frac{r_j}{w_2}, \tilde{r} = \frac{r}{w_2}$. The weight w_1 represents the fraction of the total budget allocated to input data collection, while the weight w_2 represents the fraction of the total budget allocated to generating simulation replications. Note that $\mathrm{Var}_{ij}(\theta^*) = \frac{L_1(\tilde{p}_i,\tilde{p}_j,\tilde{p})}{w_1} + \frac{L_2(\tilde{r}_i,\tilde{r}_j,\tilde{r})}{w_2}$. Note that the minimization of $L_1(\tilde{p}_i,\tilde{p}_j,\tilde{p})$ is an independent problem that does not depend on w_1, w_2 and other parameters. A similar argument holds for the minimization of $L_2(\tilde{r}_i,\tilde{r}_j,\tilde{r})$. Based on this observation, we first solve two sub-problems:

$$\text{(P1)} \quad \begin{aligned} \min_{\tilde{p}_i,\tilde{p}_j,\tilde{p}} \quad & L_1(\tilde{p}_i,\tilde{p}_j,\tilde{p}) \\ s.t. \quad & \tilde{p}_i + \tilde{p}_j + \tilde{p} = 1 \\ & \tilde{p}_i,\tilde{p}_j,\tilde{p} \geq 0 \end{aligned} \qquad \text{(P2)} \quad \begin{aligned} \min_{\tilde{r}_i,\tilde{r}_j,\tilde{r}} \quad & L_2(\tilde{r}_i,\tilde{r}_j,\tilde{r}) \\ s.t. \quad & \tilde{r}_i + \tilde{r}_j + \tilde{r} = 1 \\ & \tilde{r}_i,\tilde{r}_j,\tilde{r} \geq 0. \end{aligned}$$

For (P1), note that the problem is equivalent to (2.1) (by setting $\tilde{q}_i = \lambda_i\tilde{p}_i$, $\tilde{q}_j = \lambda_j\tilde{p}_j$ and $\tilde{q} = \lambda\tilde{p}$). Therefore, as we have discussed in Theorem 2, (P1) has closed-form optimizer, denoted as $\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*$ and optimal value $L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)$. Analogously, the problem (P2) also adopts the same structure and has closed-form optimal solution $\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*$ and optimal value $L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)$. We then consider the optimization problem

$$\begin{aligned} \min_{w_1,w_2} \quad & \frac{L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)}{w_1} + \frac{L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)}{w_2} \\ s.t. \quad & w_1 + w_2 = 1 \\ & w_1, w_2 \geq 0. \end{aligned} \tag{2.3}$$

The optimal solution of (2.3) is $w_1^* = \frac{\sqrt{L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)}}{\sqrt{L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)} + \sqrt{L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)}}$, $w_2^* = \frac{\sqrt{L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)}}{\sqrt{L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)} + \sqrt{L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)}}$. Therefore, the optimal solution for (2.2) is $p_i^* = w_1^* \tilde{p}_i^*, p_j^* = w_1^* \tilde{p}_j^*, p^* = w_1^* \tilde{p}^*, r_i^* = w_2^* \tilde{r}_i^*, r_j^* = w_2^* \tilde{r}_j^*, r^* = w_2^* \tilde{r}^*$, and the optimal value for $\text{Var}_{ij}(\theta^*)$ is $\frac{L_1(\tilde{p}_i^*, \tilde{p}_j^*, \tilde{p}^*)}{w_1^*} + \frac{L_2(\tilde{r}_i^*, \tilde{r}_j^*, \tilde{r}^*)}{w_2^*}$.

We conclude by noting that joint resource allocation problem in presence of correlation in both input data and simulation can be decoupled into three sub-problems, each admitting a closed-form solution.

2.5 Conclusion

In this chapter, we consider the target of selecting the system with better expected performance between two stochastic systems, given fixed resource budget. The resource can be used to collect more input data to reduce input uncertainty, and to implement more simulation replications to eliminate simulation error. We model the resource allocation problem as an optimization problem. The objective is appropriately allocating resource to maximize the probability of correctly selecting the best system. We exploit correlation structure of input data and common random numbers in simulation to save costs, and prove closed-form optimal resource allocation solutions. Future work may include extensions to comparison of a larger number of different systems, and numerical experiments based on real data and examples.

Appendices

2.A Omitted Proof of Section 2.2

Proof of Theorem 1. Since all third-order partial derivatives of the log-likelihood functions are uniformly bounded, we have

$$\sum_{j=1}^{\tilde{N}(pc)} \nabla \tilde{L}_{ij}(\hat{\theta}_i) + \sum_{j=1}^{N_i(p_i c)} \nabla L_{ij}(\hat{\theta}_i) = 0.$$

The mean value theorem implies that

$$\sqrt{c}(\hat{\theta}_i - \theta_i^*)\Big(\frac{\tilde{N}(pc)}{c}H_i + \frac{N_i(p_i c)}{c}H_i + o_p(1)\Big) = -\frac{\sum_{j=1}^{\tilde{N}(pc)} \nabla \tilde{L}_{ij}(\theta_i^*)}{\sqrt{c}} - \frac{\sum_{j=1}^{N_i(p_i c)} \nabla L_{ij}(\theta_i^*)}{\sqrt{c}},$$

where the notion $o_P(1)$ indicates a small random quantity that weakly converges to zero as $c \to \infty$, and Hessian matrix H_i is given by

$$H_i = \big(\mathbb{E}\,\frac{\partial^2}{\partial\theta_k \partial\theta_l} L_{i1}(\theta^*) : 1 \le k \le l \le d_i\big).$$

Since $\tilde{N}(\cdot)$ and $N_i(\cdot)$ are renewal processes, we have

$$\frac{\tilde{N}(pc)}{c} + \frac{N_i(p_i c)}{c} \xrightarrow{a.s.} \lambda p + \lambda_i p_i,$$

hence

$$c^{\frac{1}{2}}(\hat{\theta}_i - \theta_i^*) = -\frac{1}{\lambda p + \lambda_i p_i}\Big(\sum_{j=1}^{\tilde{N}(pc)} \frac{\nabla \tilde{L}_{ij}(\theta^*) H_i^{-1}}{\sqrt{c}} + \sum_{j=1}^{N_i(p_i c)} \frac{\nabla L_{ij}(\theta^*) H_i^{-1}}{\sqrt{c}}\Big) + o_p(1).$$

With respect to j, each $\nabla \tilde{L}_{ij}(\theta^*) H_i^{-1}$ and $\nabla L_{ij}(\theta^*) H_i^{-1}$ are i.i.d. and we apply central limit theorem. Therefore, when $c \to \infty$, we have

$$c^{\frac{1}{2}}((\hat{\theta}_i - \theta_i^*) : i \in [m]) \Rightarrow \Big(-\frac{1}{\lambda p + \lambda_i p_i}\big(\sqrt{\lambda p}\,\tilde{G}_i + \sqrt{\lambda_i p_i}\,G_i\big) : i \in [m]\Big).$$

□

2.B Omitted Proof of Section 2.3

Proof of Theorem 2. For problem (2.1), let $\mathcal{P} \subset \mathcal{S}$ be the set of globally optimal solutions. Denote $s_i := \frac{1}{\lambda_i} = \mathbb{E}\,\tau_{i1}$, $s_j := \frac{1}{\lambda_j} = \mathbb{E}\,\tau_{j1}$, $s := \frac{1}{\lambda} = \mathbb{E}\,\tilde{\tau}_1$ and $v(q_i, q_j, q) := \frac{\sigma_i^2}{q_i} + \frac{\sigma_j^2}{q_j} - \frac{2q\rho_{ij}\sigma_i\sigma_j}{(q+q_i)(q+q_j)}$. Because all the constraints of problem (2.1) are linear in q_i, q_j, q, if $(q_i, q_j, q)^\top \in \mathcal{P}$, it satisfies the following Karush–Kuhn–Tucker (KKT) conditions (see, for example, Lemma 5.1.4 from [6]):

$$
\begin{aligned}
&s_iq_i + s_jq_j + sq = 1,\\
&\tfrac{\partial v}{\partial q_i} - \mu_i + s_iu = 0,\\
&\tfrac{\partial v}{\partial q_j} - \mu_j + s_ju = 0,\\
&\tfrac{\partial v}{\partial q} - \mu + su = 0,\\
&\mu_iq_i = \mu_jq_j = \mu q = 0,\\
&\mu_i, \mu_j, \mu \geq 0,
\end{aligned}
\tag{2.4}
$$

where μ_i, μ_j, μ, u are KKT multipliers. The KKT condition describes a necessary condition for the optimality of a feasible point, for which the gradient of the objective function at the feasible point should be orthogonal to the feasible set $\mathcal{S}$. Suppose that $(q_i^*, q_j^*, q^*)^\top \in \mathcal{P}$ satisfies $q_i^*, q_j^*, q^* > 0$. Then, according to (2.4), KKT multipliers μ_i, μ_j and μ are equal to 0. Thus, we have

$$
\begin{aligned}
&\Big(\frac{\partial}{\partial q_i}v(q_i^*, q_j^*, q^*), \frac{\partial}{\partial q_j}v(q_i^*, q_j^*, q^*), \frac{\partial}{\partial q}v(q_i^*, q_j^*, q^*) - \frac{\partial}{\partial q_i}v(q_i^*, q_j^*, q^*) - \frac{\partial}{\partial q_j}v(q_i^*, q_j^*, q^*)\Big)\\
=&(-s_iu, -s_ju, (-s + s_i + s_j)u).
\end{aligned}
\tag{2.5}
$$

By multiplying both sides of (2.5) by $(q_1^* + q^*)^2(q_2^* + q^*)^2$, calculating the gradient of $v(q_i, q_j, q)$ and eliminating KKT multiplier u, we have a system of linear equations about q_i^*, q_j^*, q^*:

$$
\begin{aligned}
&s_iq_i^* + s_iq_i^* + sq^* = 1,\\
&2\rho_{ij}\sigma_i\sigma_js_i(q_i^* + q^*) + (s_i + s_j - s)\left(\sigma_i^2\left(q_j^* + q^*\right) - 2\rho_{ij}\sigma_i\sigma_jq^*\right) = 0,\\
&2\rho_{ij}\sigma_i\sigma_js_j(q_j^* + q^*) + (s_i + s_j - s)\left(\sigma_j^2\left(q_i^* + q^*\right) - 2\rho_{ij}\sigma_i\sigma_jq^*\right) = 0.
\end{aligned}
\tag{2.6}
$$

We discuss 3 different cases:

Case I: If $\rho_{ij} \neq 0$, and one of the following equations holds: ① $s_i + s_j = s$, ② $(s_i + s_j - s)\,\sigma_i^2 = 2s_j\rho_{ij}\sigma_i\sigma_j$, ③ $(s_i + s_j - s)\,\sigma_j^2 = 2s_i\rho_{ij}\sigma_i\sigma_j$, then (2.6) has no solutions. Therefore, there does not exist a KKT point that has three positive elements.

Case II: If $\rho_{ij} \neq 0$ and none of ①, ②, ③ hold, then (2.6) has a unique solution. We show that this solution is not optimal for (2.1), which contradicts the optimality of $(q_i^*, q_j^*, q^*)^\top$. Let $d = (d_i, d_j, d_{ij})^\top$ be a feasible direction such that $s_id_i + s_jd_j + sd_{ij} = 0$, and there exists $\epsilon_0 > 0$ enough small such that $(q_i^*, q_j^*, q^*)^\top + \epsilon_0 d$ is still feasible. Denote the value, gradient and Hessian matrix of v at $(q_i^*, q_j^*, q^*)^\top$ to be respectively v^*, g^* and Q^*. We have

$$
v\left(q_i^* + \epsilon d_i, q_j^* + \epsilon d_j, q^* + \epsilon d_{ij}\right) = v^* + \epsilon d^\top g^* + \frac{1}{2}\epsilon^2 d^\top Q^* d + o\left(||\epsilon d||^2\right).
$$

The fact that $(q_i^*, q_j^*, q^*)^\top$ is a KKT point ensures that $d^\top g^* = 0$. We have the following subcases:

(i) $s_i+s_j > s$, $\rho_{ij} < 0$ or $s_i+s_j < s$, $\rho_{ij} > 0$. In this case, we choose $d = (-s, -s, s_i+s_j)^\top$, and find that

$$d^\top Q^* d = \frac{(s_i+s_j-s)\left(\sigma_j^2\left(s_i+s_j-s\right)-2\rho_{ij}\sigma_i\sigma_j s_i\right)^2\left(\sigma_i^2(s_i+s_j-s)-2\rho_{ij}\sigma_i\sigma_j s_j\right)^2}{4(\rho_{ij}\sigma_i\sigma_j)^3}.$$

Since $\frac{s_i+s_j-s}{\rho_{ij}^3} < 0$, $d^\top Q^* d < 0$, there exists $0 < \epsilon_1 < \epsilon_0$ such that $v(q_i^* + \epsilon_1 d_i, q_j^* + \epsilon_1 d_j, q^* + \epsilon_1 d_{ij}) < v^*$. Therefore $(q_i^*, q_j^*, q^*)^\top$ is not a minimal point.

(ii) $s_i + s_j < s$, $\rho_{ij} < 0$ or $s_i + s_j > s$, $\rho_{ij} > 0$. We choose $d = (-s, 0, s_i)^\top$, and calculate that

$$d^\top Q^* d = \frac{s_i(s_i-s)\left(\sigma_j^2\left(s_i+s_j-s\right)-2\rho_{ij}\sigma_i\sigma_j s_i\right)^2\left(\sigma_i^2(s_i+s_j-s)-2\rho_{ij}\sigma_i\sigma_j s_j\right)^2}{4(s_i+s_j-s)(\rho_{ij}\sigma_i\sigma_j)^3}.$$

Since we have assumed that $s_i < s$, $d^\top Q^* d < 0$. Similarly to case (i), $(q_i^*, q_j^*, q^*)^\top$ is not a minimal point.

Case III: When $\rho_{ij} = 0$, we discuss about three subcases: $s_i + s_j > s$, $s_i + s_j < s$ and $s_i + s_j = s$. If $s_i + s_j > s$, for any point $(q_i, q_j, q)^\top \in \mathcal{S}$ satisfying $q_i, q_j, q > 0$, $d = (-s, -s, s_i + s_j)^\top$ is a descent direction for both $\frac{\sigma_i^2}{q_i+q}$ and $\frac{\sigma_j^2}{q_j+q}$ as a function of (q_i, q_j, q). Therefore, there exists no optimal solution that has three positive elements. If $s_i + s_j < s$, let $d = (s, s, -s_i - s_j)^\top$, we reach the same conclusion as when $s_i + s_j > s$. If $s_i + s_j = s$, define $w_i := q_i + q$ and $w_j := q_j + q$. The problem (2.1) is then converted to

$$\begin{aligned} \min_{w_i, w_j \geq 0} \quad & \frac{\sigma_i^2}{w_i} + \frac{\sigma_j^2}{w_j} \\ s.t. \quad & s_i w_i + s_j w_j = 1. \end{aligned}$$

The optimal w_i^* and w_j^* satisfies $\frac{\sigma_i}{w_i^*\sqrt{s_i}} = \frac{\sigma_j}{w_j^*\sqrt{s_j}}$. So in this case, $(q_i, q_j, q)^\top \in \mathcal{P}$ if and only if $\frac{\sigma_i}{\sqrt{s_i}(q_i+q)} = \frac{\sigma_j}{\sqrt{s_j}(q_j+q)}$.

Summarizing the conclusions of case I, II and III, whenever $s_i + s_j \neq s$ or $\rho_{ij} \neq 0$, the optimal solution can only be achieved on the boundary of the feasible region. That is, the optimal solution(s) must have at least one element as zero. When $s_i + s_j = s$ and $\rho_{ij} = 0$, there exists an optimal solution that has at least one zero element and also an optimal solution that has all positive elements. □

2.C Omitted Proof of Section 2.4

Proof of Theorem 3. We decompose the deviation into two parts and consider characteristic function. Denote

$$Y_{1,C} = C^{\frac{1}{2}}((\hat{\alpha}_i(\hat{\theta}_i) - \alpha_i(\hat{\theta}_i)) - (\hat{\alpha}_j(\hat{\theta}_j) - \alpha_j(\hat{\theta}_j)))$$

and

$$Y_{2,C} = C^{\frac{1}{2}}((\alpha_i(\hat{\theta}_i) - \alpha_i(\theta_i^*)) - (\alpha_j(\hat{\theta}_j) - \alpha_j(\theta_j^*)))$$

For any $t \in \mathbb{R}$, we have

$$\mathbb{E}\left[e^{\mathbf{i}t\left(Y_{1,C}+Y_{2,C}\right)}\right] = \mathbb{E}\left\{\mathbb{E}\left[e^{\mathbf{i}t\left(Y_{1,C}+Y_{2,C}\right)}\left|\hat{\theta}_i, \hat{\theta}_j\right|\right\} = \mathbb{E}\left\{e^{\mathbf{i}tY_{2,C}}\mathbb{E}\left[e^{\mathbf{i}tY_{1,C}}|\hat{\theta}_i, \hat{\theta}_j\right]\right\}. \tag{2.7}$$

Under our allocation rule, according to Theorem 1,

$$Y_{2,C} \Rightarrow W_{i,j} \quad \text{as } C \to \infty. \tag{2.8}$$

Here $W_{i,j}$ is Gaussian with mean zero and variance

$$\frac{\sigma_i^2}{\lambda p + \lambda_i p_i} + \frac{\sigma_j^2}{\lambda p + \lambda_j p_j} - \frac{2\lambda p\, c_{ij}}{(\lambda p + \lambda_i p_i)(\lambda p + \lambda_j p_j)}.$$

On the other hand, conditioned on $\hat{\theta}_i$ and $\hat{\theta}_j$, as $C \to \infty$,

$$Y_{1,C} \Rightarrow V_{ij}(\hat{\theta}). \tag{2.9}$$

Conditioned on $\hat{\theta}$, the rv $V_{ij}(\hat{\theta})$ is Gaussian with mean zero and variance

$$\frac{D_i(\hat{\theta}_i)}{\mu r + \mu_i r_i} + \frac{D_j(\hat{\theta}_j)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\hat{\theta})}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)}.$$

Therefore,

$$\mathbb{E}\left[e^{\mathbf{i}tY_{1,C}}|\hat{\theta}_i, \hat{\theta}_j\right] = \exp(-\frac{t^2}{2}(\frac{D_i(\hat{\theta}_i)}{\mu r + \mu_i r_i} + \frac{D_j(\hat{\theta}_j)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\hat{\theta})}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)})) + o_p(1). \tag{2.10}$$

$D_i(\cdot), D_j(\cdot), D_{ij}(\cdot)$ are continuous functions, so we have

$$\mathbb{E}\left[e^{\mathbf{i}tY_{1,C}}|\hat{\theta}_i, \hat{\theta}_j\right] \Rightarrow \exp(-\frac{t^2}{2}(\frac{D_i(\theta_i^*)}{\mu r + \mu_i r_i} + \frac{D_j(\theta_j^*)}{\mu r + \mu_j r_j} - \frac{2\mu r\, D_{ij}(\theta^*)}{(\mu r + \mu_i r_i)(\mu r + \mu_j r_j)})) \tag{2.11}$$

as $C \to \infty$. Finally, we induce that

$$\mathbb{E}\left[e^{\mathbf{i}t\left(Y_{1,C}+Y_{2,C}\right)}\right] \Rightarrow \exp(-\frac{t^2}{2}\operatorname{Var}_{ij}(\theta^*))$$

as $C \to \infty$, which implies the conclusion in Theorem 3. □

Chapter 3

Simulation Optimization via Multi-Resolution System Approximations

In this chapter, we focus on the specific class of simulation-optimization problems with continuous multi-dimensional decision variable and convex feasible region, for complicated stochastic systems. For complicated stochastic systems, the exact simulation of system performance can be expensive or even take infinite computation time in expectation. For example, the simulation of a service system operation can be time-consuming when the demand volume is huge, the time horizon is long, or the operations logic is complex. As another example, the solution of a general multi-dimensional stochastic differential equation (SDE) may arise as a stochastic object in applications of financial systems and queueing systems, the exact simulation of which may not get executed in finite computation time in expectation. Therefore, given the complicated system logic and stochastic objects, the exact simulation of the objective function at a certain value of decision variable is computationally expensive or impossible. The associated optimization problems are even more challenging, since the difficult or impossible exact simulations to evaluate the objective function need to be done on multiple values of the decision variable to find a good solution.

For the complicated stochastic system in consideration, referred to as the *original system*, at some times a sequence of approximations can be constructed, with finer and finer approximation resolution. We presume in mind that the expected simulation time to evaluate the performance for the original system is infinite. The expected simulation time to evaluate the approximations, on the other hand, is finite, though it can be arbitrarily large depending on the approximation resolution. The approximation of the original system can either be applied to the simulation logic or the random object, or both. We refer to an approximation of the original system as an *approximating system*. For example, for stochastic systems involving complicated continuous-time stochastic uncertainties in operations, a popular way to construct approximating systems is through time discretization at different resolutions. We assume that the constructed approximating systems present finite simulation time to evaluate, as an advantage compared to the original system. However, the approximation often inevitably incurs a bias compared to the original system. In general, the sequence of approximations can be constructed in a way that they eventually converge to the original system and the bias vanishes in the limit. As a result, such sequence of approximating systems incurs smaller and smaller bias, while incurring larger and larger simulation cost. Our work focuses on strategically using the sequence of approximating systems to design and analyze gradient-based simulation-optimization algorithms that solve optimization problems associated with the original system.

Gradient-based algorithms have proved to be effective for a large number of simulation optimization problems, especially when the decision variable is high-dimensional. However, when the original stochastic system is too complicated or impossible to exactly simulate in finite time, the computation of gradient estimator associated with the original system shares the same challenge, if not more. Our focus, therefore, is to utilize the sequence of approximating systems, each of which can be simulated in finite time at finite cost, to construct gradient-based simulation-optimization algorithms. As the approximation resolution increases, an approximating system becomes closer to the original system (i.e. higher resolution), but this higher resolution comes at a price with an increasing simulation cost. We leverage the idea that lower-resolution approximating systems can be used to evaluate gradient estimators at earlier stages in a gradient-based optimization algorithm. The intuition is that at early stages, the algorithm may start with a choice of decision variable that is likely far away from the optimal solution, and there is no need to use a high-resolution system to construct the gradient estimator that comes with a high simulation cost. The algorithm updates the value of decision variable by iterations. As the algorithm proceeds, the value of decision variable is presumed to be closer to the optimum. Approximating systems with higher and higher resolution are used to construct gradient estimators, which are then used to update the decision variable.

The computation of gradient estimators for approximating systems has its own challenges. The challenges often comes from discontinuity. Such discontinuity may come from time discretization or piecewise constant approximations, which are often used to construct approximating systems. When there is no continuity on the objective function associated with an approximating system, standard gradient estimators such as Infinitesimal Perturbation Analysis (IPA)/Automatic Differentiation (AD)/Backpropagation (BP) gradient estimators may incur an uncontrollable bias or even take an opposite sign compared to the true gradient. In this case, the gradient-based algorithm with IPA/AD/BP gradient estimators can fail badly and diverge. See [24] for an example. To circumvent the discontinuity challenge and to avoid the potentially difficult likelihood evaluation for complicated systems, we use the finite difference (FD) method with a stochastic direction to construct gradient estimators for an approximating system. The construction of gradient estimators via finite difference method is based on observations of only stochastic realizations of the objective function at given values of decision variables. In the scenarios of discontinuity, the use of finite difference gradient estimators, despite its sometimes unpleasant dependence on the dimension d, can at least guarantee the control of bias and therefore the optimization algorithm's convergence. Under the assumption that the expected performance of the original system is strongly convex and smooth, we prove that the associated gradient-based algorithm converges to the optimum without any assumption on convexity or continuity with the approximating systems. The required computation cost to achieve a given precision level is proved to have a moderate polynomial order of dependence on the dimension of decision variable.

Further, we develop and analyze gradient-based simulation-optimization algorithms, for which the gradient estimator used in any single step is constructed by coupling multiple approximating systems at different resolutions, instead of just using one approximating system. An effective framework is the multilevel regime introduced by [48] and [36]. The core idea of multilevel regime is to intelligently integrate simulations at different resolutions to improve computational efficiency, when the approximating systems at different resolutions can naturally be coupled. Under the classical multilevel regime ([36]), we construct multilevel FD gradient estimators (with stochastic directions) to support gradient-based optimization algorithms. At each step, a multilevel gradient estimator utilizes multiple coupled approximating systems at different resolutions. We prove that the use of multilevel gradient estimators can improve the computational efficiency of the associated simulation-optimization algorithms, by reducing the order of computation cost required to achieve an arbitrary level of precision, compared with algorithms that use single-level gradient estimators. In particular, when the multilevel FD gradient estimator is used, we investigate the algorithm performance in terms of its dependence on the decision space dimension, and prove a moderate order polynomial dependence as an upper bound.

Our contributions to the literature are summarized as follows.

- We present a framework to optimize a stochastic system that is too complicated to exactly simulate, with the aid of approximating systems. The sequence of approximations has finer and finer approximation resolution but higher and higher cost to simulate. We propose a new gradient-based simulation-optimization algorithm that sequentially utilizes approximating systems of higher and higher resolutions to construct gradient estimators and do gradient search.

- Under the assumption that the objective function of the original system is strongly convex and smooth, while allowing objective functions associated with approximating systems to be nonconvex and discontinuous, we prove algorithm convergence, convergence rates, and optimal algorithm design for the simulation-optimization algorithms with finite difference estimators. When the finite difference gradient estimators are used, we demonstrate the dependence on the dimension for the algorithm performance and optimal parameters choices.

- When the sequence of approximating systems can be coupled, we propose new multilevel gradient-based simulation-optimization algorithms. We prove that the use of multilevel gradient estimators can improve the computational efficiency of the associated simulation-optimization algorithms, by reducing the order of computation cost required to achieve an arbitrary level of precision, compared with algorithms that use single-level gradient estimators.

Built on contents described above on algorithm design and theoretical guarantees, we describe in the following a more summarized message from this work. This work considers situations where one wants to use simulation to optimize a complicated stochastic system where the system is difficult to simulate but instead has a sequence of simulatable approximating systems. Instead of the baseline method by simply using a single approximating system, this work more carefully integrates approximating systems of different resolutions to design optimization algorithms. If we count the expected simulation cost to achieve the same level of optimization accuracy, say a small $\epsilon > 0$, the algorithm proposed in this work can enjoy a lower cost relative to the baseline, by a factor as polynomials of $1/\epsilon$.

3.1 Related Work

Our work is related to the literature of gradient-based simulation optimization. We categorize the related literature in Table 3.1 for reference.

In the scenarios where the stochastic objective functions contain discontinuity, there may be alternative gradient estimators other than FD gradient estimator that have controllable bias or no bias. The use of these alternative gradient estimators can have their own challenges. Smoothed perturbation analysis (SPA) can circumvent the bias issue by using a conditioning technique ([40]), but a good choice of conditioning is problem-dependent. Another gradient estimation technique, likelihood ratio or the score function method ([39, 85]), does not directly apply for structural parameters, which may limit its applicability in general. Measure-valued differentiation ([46, 47]) requires structural information of the dynamics of the simulation systems. In Peng et al. [83], a generalized likelihood ratio (GLR) method was proposed, the use of which in our context would require each random input for the approximating systems to admit a density, which sometimes can be violated. Therefore, in this work, in presence of discontinuity, we choose the use of finite difference gradient estimator, to accommodate relatively more general structures for the approximating systems. Moreover, the use of finite difference gradient estimators can provide theoretical guarantee for the convergence of gradient-based algorithms even when the objective functions associated with the approximating systems are not convex or smooth.

Before our work, the use of multilevel regime appeared in stochastic approximation, by [27], [21] and [20]. Their works do not consider constructions of gradient estimators in presence of discontinuity or the algorithm performance's dependence on the dimension from a computational aspect, which is an important aspect in many simulation-optimization problems with high-dimensional decision space. Blanchet, Glynn, and Pei [8] propose multilevel sample-average approximation algorithms for stochastic convex optimization. Their work applies the multilevel technique to strategically use the n independent and identically distributed samples for the original system to improve convergence rates and/or derive unbiased estimators for the optimal value. Their use of multilevel technique utilizes an increasing

Gradient-based simulation optimization		
Exact simulation for the original system or its gradient oracle is available		
First-order gradient estimator	Zeroth-order gradient estimator	
	Simultaneous perturbation gradient estimator	Random direction gradient estimator
[70],[35], [28],[96], [24]	[50],[93], [28]	[35], [22], [78]
Exact simulation for the original system or its gradient oracle is unavailable		
[110], [44], [71], [56],[105], [27], [20]		

Table 3.1: A table for summarizing existing gradient-based simulation optimization algorithms. We categorize the related literature of gradient-based simulation optimization by whether they adopt the assumption that exact simulation for the original system or its gradient oracle is available, or not.

number of samples for the system but our use of multilevel technique alternatively utilizes increasing approximating resolution for the system. These two perspectives are orthogonal and can potentially be integrated.

The idea of using different resolutions of approximations to study complicated stochastic systems has also been considered by a line of work on multi-fidelity simulation, but with different objectives from ours. Xu et al. [105] discuss budget allocation strategies for the problem of ranking and selection using a high-fidelity approximation and a low fidelity approximation. Song et al. [92] extends the framework of [105] and proposes an optimal sampling policy that uses a low-fidelity model to improve the computational efficiency of ranking and selection. Balabanov and Venter [3] considers the use of one-dimensional gradient-based optimization in a high-fidelity model while using the low-fidelity model to do finite difference gradient calculations. Huang et al. [56] discusses the effective use of low-fidelity model to enhance surrogate models such as kriging.

In the literature, there has been an increasing attention on using biased gradient estimators to solve stochastic optimization problems, for which the bias is rather intrinsic but not caused by approximations to optimize systems that are impossible to exactly simulate. Some most recent work includes [24, 54, 1, 98, 113, 112]. In our work, we overcome the issues of bias by dynamically using approximating systems at higher and higher resolutions. Some relevant pioneering work include [110], [44] and [71].

3.2 Problem Setting

Consider a complex stochastic system in which a decision needs to be made to optimize an objective function that involves an expected system performance. We denote $\theta \in \Theta \subset \mathbb{R}^d$ as the decision variable, in which Θ is a closed and convex subset of $\mathbb{R}^d$. Fix a probability space $(\Omega, \mathcal{F}, P)$ that supports the stochastic system. We summarize all the stochastic uncertainties involved in the stochastic system as a random object Y that takes value in a Polish space $\mathcal{Y}$. This random object Y, for example, may include non-stationary doubly stochastic Poisson processes or solutions of stochastic differential equations, depending on the system in consideration. The system performance is described as a product-measurable function $G : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$ that maps the decision variable $\theta \in \mathbb{R}^d$ and the realization of the random object Y into a real-valued scalar. The function G may involve various system operations logic, system performance and operational costs. For complex stochastic systems in consideration, this performance function G does not have closed-form representation and can only be evaluated by simulation. To obtain the optimal decision, the following simulation-optimization problem needs to be solved.

$$\min_{\theta \in \Theta} g(\theta) := \mathbb{E}[G(\theta, Y)]. \tag{3.1}$$

We assume that $\mathbb{E}\, G(\theta, Y)^2 < \infty$ throughout this chapter. The random object Y and the performance function G are the key components of a simulation task, which are respectively referred to as *random input* and *logic*. For a complicated stochastic system, both components can incur significant simulation cost in the evaluation of the expected performance $\mathbb{E}[G(\theta, Y)]$. For example, the function G may involve the logic and operational rule of a complicated stochastic system. There are other scenarios where G is simple but the random object Y is difficult or even impossible to exactly simulate. For example, the random object Y can be a stochastic process $Y = (Y(t) : t \geq 0)$ as a solution of a general multi-dimensional stochastic differential equation (SDE). For a general SDE, Y is viewed as an infinite dimensional object that cannot be exactly simulated in finite computing time.

In complicated stochastic systems where $G(\theta, Y)$ is difficult or impossible to simulate, we presume the simulation cost for $G(\theta, Y)$ to be infinity and assume that we have the ability to construct a sequence of approximations. In general, we assume that there exists a sequence of systems $(G_k(\theta, Y_k) : k \geq 0)$ that approximates $G(\theta, Y)$ with finer and finer resolution. The expectation of $G_k(\theta, Y_k)$ is denoted as $g_k(\theta)$. The sequence $(G_k(\theta, Y_k) : k \geq 0)$ approximates $G(\theta, Y)$ in the sense that for any decision variable θ, the asymptotic approximation bias diminishes,

$$\lim_{k \to \infty} \mathbb{E}(G_k(\theta, Y_k) - G(\theta, Y)) = 0. \tag{3.2}$$

Or, sometimes the approximation can be constructed to adopt a stronger convergence, given by

$$\lim_{k \to \infty} \mathbb{E}(G_k(\theta, Y_k) - G(\theta, Y))^2 = 0. \tag{3.3}$$

We will specify case by case which approximation is needed in the theoretical results to be provided. For any given k, the cost to simulate a copy $G_k(\theta, Y_k)$ is finite and increasing with

k. The approximation $G_k(\theta, Y_k)$ may involve an approximation on the system operations logic G, noted as G_k, or an approximation on the random object Y, noted as Y_k, or both. If approximations are done for the complicated random object Y, then the approximating sequence $\{Y_k : k \geq 1\}$ usually involve time or space discretizations of Y. Our goal is to design and analyze simulation-optimization algorithms that solves the original optimization problem (3.1) up to any given precision level, through an effective use of the approximating systems.

For the original stochastic system, if $G(\theta, Y)$ were easy to simulate, then with an initial guess θ_0, a gradient-based algorithm could be derived, generally taking the form of

$$\theta_t = \theta_{t-1} - \gamma_t \hat{\nabla}_\theta g(\theta_{t-1}), \tag{3.4}$$

in which $\hat{\nabla}_\theta g(\theta_{t-1})$ denotes estimated value of gradient of $g(\cdot)$ at point θ_{t-1}, constructed from the simulation of $G(\theta, Y)$, and $\gamma_t \in \mathbb{R}^+$ is a carefully chosen step size. However, when the original system $G(\theta, Y)$ is too complicated and impossible to exactly simulate, the associated gradient estimators are often not available at finite cost. Our focus, therefore, is to utilize the sequence of approximating systems, each of which can be simulated at finite cost, to construct gradient-based simulation-optimization algorithms. When k becomes larger, the k-th approximating system becomes closer to the original system, but this higher resolution comes at a price with an increasing computation cost. Our work leverages the thought that lower-resolution approximating systems can be used to compute gradient estimators at earlier stages in a gradient-based algorithm. As the algorithm proceeds, higher-resolution approximating systems are used to derive gradient estimators, so that eventually the algorithm can achieve any given level of precision. The details will be discussed in the following sections.

3.3 Simulation-Optimization Algorithms with Finite Difference Gradient Estimators

In this section, we propose gradient-based simulation-optimization algorithms that sequentially utilize gradients from approximating systems with increasing resolution. To facilitate the theoretical analysis for the proposed algorithms, we presume the original objective function g to be strongly convex and contains a unique local optimal solution θ^*, which is also a global optimal solution. When the objective function g is not strongly convex, the algorithms and analysis can be extended to find strongly convex local optimums. Formal assumptions will be stated later in this section. Despite the convexity assumption on g, we make no convexity or continuity assumptions on the approximating systems g_k's. One specific reason is that many approximations of a complicated infinite-dimensional stochastic system are done through discretization in sample paths and/or objective functions, naturally leading to discontinuity and non-convexity in g_k.

When there is no regularity conditions such as continuity on the approximating function $G_k(\theta, Y_k)$ or $g_k(\theta)$, standard gradient estimators to evaluate $\partial g_k(\theta)/\partial\theta$ such as Infinitesimal

Perturbation Analysis (IPA)/Automatic Differentiation (AD) gradient estimators may incur an uncontrollable bias or even take an opposite sign compared to the true gradient; see [24]. Such uncontrollable biases may lead gradient-based algorithms to diverge; see [24] and [1] for example. In the scenarios of discontinuity, the use of finite difference gradient estimators, despite its sometimes unpleasant dependence on the dimension d, can at least guarantee the control of bias and therefore the optimization algorithm's convergence. In presence of discontinuity, we propose simulation optimization algorithms based on finite difference gradient estimators and provide theoretical guarantee for the convergence of the algorithms.

Construction of FD Gradient Estimator using Approximating Systems

To construct a FD gradient estimator, instead of computing the function value perturbation in each of the d dimension, we consider the FD gradient estimator with a randomly drawn direction; see [68] and [23] for discussions on this type of gradient estimator. Consider a non-increasing sequence of positive parameters $\{h_k\}_{k=1}^{\infty}$, a distribution ν on $\mathbb{R}^d$, and a random vector Z sampled from ν. The distribution ν satisfies $\mathbb{E}_\nu[ZZ^\top] = I$. For the approximating system $G_k(\theta, Y_k)$ with index k, the FD gradient estimator is given by

$$\frac{G_k(\theta + h_k Z_k, Y_k) - G_k(\theta, Y_k)}{h_k} Z_k. \tag{3.5}$$

The estimator (3.5) is motivated by the following fact: If $\nabla_\theta g_k(\theta)$ exists and $\|\nabla_\theta g_k(\theta)\|$ is bounded by a constant for every $\theta \in \Theta$,

$$\begin{aligned} &\lim_{h\to 0} \mathbb{E}_\nu \left[\frac{G_k(\theta + hZ, Y_k) - G_k(\theta, Y_k)}{h} Z\right] \\ =&\lim_{h\to 0} \mathbb{E}_\nu \left[\frac{g_k(\theta + hZ) - g_k(\theta)}{h} Z\right] \\ =&\mathbb{E}_\nu[< \nabla_\theta g_k(\theta), Z > Z] = \nabla_\theta g_k(\theta). \end{aligned} \tag{3.6}$$

Consequently, if h_k is sufficiently small, (3.5) should be a nearly unbiased estimator of the gradient $\nabla_\theta g_k(\theta)$. The property (3.6) stands even if the stochastic performance function $G_k(\cdot, Y_k)$ is not continuous. Therefore, $G_k(\cdot, Y_k)$ does not need to be continuous in θ to ensure asymptotic unbiasedness when h goes to zero.

Algorithm and Assumptions

We propose an algorithm to solve for the optimal solution that is based on the Stochastic Gradient Descent routine and uses FD gradient estimators constructed from the approximating systems. Consider an increasing positive integer sequence $\{m_t : t \geq 1\}$ and a non-increasing sequence of positive smoothing parameters $\{h_t : t \geq 1\}$. At the t-th step, the algorithm updates θ_t from the previous step by using a finite difference gradient estimator constructed from the m_t-th approximating system. Specifically, the algorithm generates N_t independent copies of Y_{m_t}, noted as $\{Y_{m_t,l}\}_{l=1}^{N_t}$ and sample independent vectors $\{Z_{t,l}\}_{l=1}^{N_t}$ from ν. Also, the algorithm runs $2N_t$ independent simulation copies of the simulation logic of the m_t-th system and obtains the gradient estimator

$$H_t(\theta) = \frac{1}{N_t} \sum_{l=1}^{N_t} \frac{G_{m_t}(\theta + h_t Z_{t,l}, Y_{m_t,l}) - G_{m_t}(\theta, Y_{m_t,l})}{h_t} Z_{t,l}. \tag{3.7}$$

The algorithm then updates θ_t as

$$\theta_{t+1} = \mathrm{pr}_\Theta(\theta_t - \gamma_t H_t(\theta_t)), \tag{3.8}$$

where pr_Θ is the orthogonal projection on Θ under Euclidean distance. γ_t is a positive scalar to be specified, which is also known as *step size*. The fact that m_t increases with t illustrates that the algorithm uses an approximating system with higher and higher resolution as it proceeds. In the beginning, the algorithm updates the optimizer relying on gradient estimators constructed from approximating systems with low resolution. When the algorithm proceeds as t increases, the algorithm uses higher-resolution simulation systems to construct the gradient estimator. Eventually, the algorithm is expected to ensure the convergence of θ_t to θ^*.

We will provide answers to the following questions:

- When does the algorithm ensure convergence of θ_t to θ^*?
- What is the convergence rate of θ_t and the limiting distribution of θ_t?
- How does the cumulative computation cost depend on the dimension d?
- How can we set algorithm parameters γ_t, h_t and N_t to save cumulative computation cost?

Before proceeding into the algorithm analysis, we make some assumptions and specifications on the original system, the sequence of approximating systems, and the computation cost of simulating a copy of the approximating system at different resolution. To focus our attention on discussing how the sequence of approximating systems are strategically used to optimize the original system, we assume that the expected performance of the original system is nicely behaved, as follows.

Assumption 1. *The expected performance $g(\theta)$ as a function of θ is L-smooth and μ-strongly convex. That is, there exists $L > 0$ such that $\|\nabla_\theta g(\theta_1) - \nabla_\theta g(\theta_2)\| \le L\|\theta_1 - \theta_2\|$ $\forall \theta_1, \theta_2 \in \Theta$. There exists $\mu > 0$ such that $(\nabla_\theta g(\theta_1) - \nabla_\theta g(\theta_2))^\top (\theta_1 - \theta_2) \ge \mu \|\theta_1 - \theta_2\|^2$ $\forall \theta_1, \theta_2 \in \Theta$.*

Note that this assumption is only imposed on the expected performance of the original system $g(\cdot) = \mathbb{E}\, G(\cdot, Y)$. The stochastic performance function $G(\theta, Y)$ can be discontinuous and non-convex in θ for some given realizations of the random object Y. In addition, for the approximating systems, the expected performance function $g_k(\cdot) = \mathbb{E}\, G_k(\cdot, Y_k)$ and the stochastic performance function $G_k(\cdot, Y_k)$ are not required to be continuous or convex.

We impose the following assumption to quantify the difference of expected performance function between the k-th approximating system and the original system:

Assumption 2. *There exists a constant $C > 0$ such that for all $k \in \mathbb{N}$ and $\theta \in \Theta$, $|\,\mathbb{E}[G_k(\theta, Y_k) - G(\theta, Y)]| = |g_k(\theta) - g(\theta)| \leq Ck^{-1}$.*

Assumption 2 gauges the resolution of the k-th approximating system. A larger k indicates a higher resolution and closer approximation.

When the smoothing parameter $h > 0$, the FD gradient estimator naturally introduces a bias. This bias depends on both h and the dimension d. For different systems and different function properties on $G_k(\theta, Y_k)$, the dependence on h may be different. A realistic assumption that allows different dependence on h is given as below.

Assumption 3. *There exist constants $\tilde{C} > 0, h_0 > 0$ and $q \geq 0$, such that for all $k \in \mathbb{N}$ and $0 < h \leq h_0$, $\mathbb{E}[\|(G_k(\theta + hZ, Y_k) - G_k(\theta, Y_k))Z\|^2] \leq \tilde{C}d^2h^q$.*

The value of q represents the dependence on h for $\mathbb{E}[\|(G_k(\theta + hZ, Y_k) - G_k(\theta, Y_k))Z\|^2]$. In general, the smoother the stochastic performance function is, the larger the q is. If the stochastic performance function $G_k(\cdot, Y_k)$ is $\Psi_k(Y_k)$-Lipschitz continuous and $\sup_{k\geq 1} \mathbb{E}[|\Psi_k(Y_k)|^2] < \infty$, we have $q = 2$ with details to be given shortly. When the continuity fails, the value of q can be as small as 0, see [2]. [71] propose additional conditions on the discontinuous performance function such that $q = 1$. In general, we allow q to be a flexible parameter, but aim to be able to deal with the most general stochastic performance function (without continuity requirement) that leads to $q = 0$.

Specifically, when $G_k(\cdot, Y_k)$ is $\Psi_k(Y_k)$-Lipschitz continuous and $\sup_{k\geq 1} \mathbb{E}[|\Psi_k(Y_k)|^2] < \infty$, we have

$$\begin{aligned} &\mathbb{E}[\|(G_k(\theta + hZ, Y_k) - G_k(\theta, Y_k))Z\|^2] \\ =&\,\mathbb{E}[\mathbb{E}[\|G_k(\theta + hZ, Y_k) - G_k(\theta, Y_k)\|^2|Z]\|Z\|^2] \\ \leq&\,\mathbb{E}[\mathbb{E}[h^2|\Psi_k(Y_k)|^2\|Z\|^2|Z]\|Z\|^2] \\ \leq&\,h^2 \sup_{k\geq 1} \mathbb{E}[|\Psi_k(Y_k)|^2]\,\mathbb{E}[\|Z\|^4] = O(d^2h^2), \end{aligned} \tag{3.9}$$

showing the notion that $q = 2$ under continuity.

In addition to the condition on the random direction $\mathbb{E}[ZZ^\top] = I$, we require the following properties on the distribution ν:

Assumption 4. *For a random vector Z sampled from ν and any $k \in \mathbb{N}$, there is a constant c_k (dependent only on k) such that $\mathbb{E}\left[\|Z\|^k\right] \leq c_k d^{\frac{k}{2}}$.*

Assumption 4 gives a bound on the moments of vector Z. For example, Assumption 4 is satisfied if Z is distributed as $N(0, I_{d\times d})$, or uniformly on the l_2-sphere of radius $\sqrt{d}$.

Convergence, Optimality, Cost, and Central Limit Theorem

In this subsection we state the main convergence result that provides a theoretical guarantee that the algorithm utilizing FD gradient estimators will eventually find θ^*. For this simulation-optimization algorithm, in each step there are four algorithm parameters m_t (label of system of use in the t-th iteration), N_t (number of independent copies of simulation to run in the t-th iteration), smoothing parameter h_t, and step size γ_t (i.e., step size in the t-th iteration) to be specified. For a set of parameters $\gamma_0, N_0, m_0 > 0$, $\beta, r, \rho \geq 0$, we consider algorithm parameters given by

$$m_t = \lceil m_0 d^2\, t^{2\rho} \rceil,\ N_t = \left\lceil N_0 d^{5-\frac{3}{2}q} t^{r+\rho(2-q)} \right\rceil,\ h_t = \frac{1}{d^{\frac{3}{2}} t^{\rho}},\ \gamma_t = \gamma_0 \frac{1}{t^{\beta}}. \tag{3.10}$$

Given the algorithm parameters, note that the estimator of θ^* after the t-th iteration is denoted as θ_t, specified by (3.7) and (3.8). We present a theorem on the properties of θ_t.

Theorem 4. *Suppose that $\beta \in (\frac{1}{2}, 1]$, $r \geq 0$, $\rho, \gamma_0, N_0, m_0 \in (0, \infty)$. If $\beta = 1$, suppose additionally that $\gamma_0 \in (\max\{\frac{2\rho}{\mu}, \frac{1+r}{\mu}\}, \infty)$. Under Assumption 1, 2, 3 and 4, there exists a $\kappa \in (0, \infty)$ such that for all $t \in \mathbb{N}$,*

$$\mathbb{E}\left[g(\theta_t) - g(\theta^*)\right] \leq \frac{1}{2} L\kappa\, t^{-(2\rho)\wedge(\beta+r)}, \tag{3.11}$$

and

$$\mathbb{E}\left[\|\theta_t - \theta^*\|^2\right] \leq \kappa\, t^{-(2\rho)\wedge(\beta+r)}, \tag{3.12}$$

where κ only depends on $\beta, r, \rho, \gamma_0, m_0, N_0, L$ and q.

Remark 5. *We remark that the dimension d only exists in the algorithm parameter m_t, N_t and h_t, while does not appear in the upper bound of the convergence rate nor in the constant κ.*

Remark 6. *The parameter ρ and r reflect the conditional bias and variance of the gradient estimator $H_t(\theta_t)$ at time step t respectively. Specifically, the conditional bias of $H_t(\theta_t)$, denoted by $B_t = \mathbb{E}[H_t(\theta_t) - \nabla g(\theta_t)]$, has the scale of $O(t^{-\rho})$. The parameter ρ controls the decreasing rate of the bias by controlling algorithm parameter m_t and h_t. As in (3.10), if we set a larger ρ, then in each time step, we use an approximating system with a finer resolution as well as a smaller smoothing parameter h_t. Therefore, with a larger parameter ρ, the bias of the FD gradient estimator from $\nabla g(\theta_t)$ is lower. The parameter r controls the conditional variance of $H_t(\theta_t)$ by controlling algorithm parameter N_t. The conditional variance of $H_t(\theta_t)$ given θ_t, has the scale of $O(t^{-r})$. The details of the scale of conditional bias and variance of $H_t(\theta_t)$ are given in the proof of Theorem 4 in the appendices.*

Remark 7. *When the objective function g is L-smooth but not strongly convex on Θ, the algorithm and analysis can be extended to find strongly convex local optimums. Suppose that $g(\cdot)$ is strongly convex on $\Theta' \subset \Theta$. By changing the iteration rule (3.8) to $\theta_{t+1} =$*

$pr_{\Theta'}(\theta_t - \gamma_t H_t(\theta_t))$*, the algorithm will ensure the convergence of* θ_t *to the minimal point in* Θ' *with the same convergence rate of Theorem 4.*

Note that this convergence rate result appears to not depend on the dimension d, because the computational cost spent at each iteration has not yet been taken into consideration. We next discuss a result on the expected computation cost associated with the algorithm for different choices of the algorithm parameters ρ, r and β, which will support the selection of the best choice of algorithm parameters to minimize the order of computation cost to achieve a given precision level. Define C_k as the computation cost for generating a single simulation run of $G_k(\theta, Y_k)$. To reflect the notion that the original system is impossible to exactly simulate, the cost increases to infinity as $k \to \infty$. This computation cost C_k can be a random variable itself and we make the following assumptions that characterize the growth rate of the expected cost when $k \to \infty$. The growth rate can either be polynomial or exponential at different rates.

Assumption 5. *As* $k \to \infty$*, the sequence of costs* $\{C_k : k \geq 1\}$ *satisfies either of the following two conditions.*

(i) $k^{-p}\,\mathbb{E}[C_k] \to \kappa_1$ *for some* $p > 0$ *and* $\kappa_1 > 0$.

(ii) $\alpha^{-k}\,\mathbb{E}[C_k] \to \kappa_2$ *for some* $\alpha > 1$ *and* $\kappa_2 > 0$.

Before proceeding into the theorem, for an algorithm with algorithm parameters specified by (3.10), we define

$$\tau(\epsilon) := \inf\{t \in \mathbb{N} : (\mathbb{E}\left[\|\theta_t - \theta^*\|^2\right])^{\frac{1}{2}} \leq \epsilon\} \tag{3.13}$$

as the number of iterations it takes for θ_t to satisfy a precision level ϵ. We can then compute the cumulative computation cost for the algorithm by the $\tau(\epsilon)$-th iteration, denoted as

$$T_{\tau(\epsilon)} = 2\sum_{j=1}^{\tau(\epsilon)} N_j C_{m_j}. \tag{3.14}$$

Theorem 8. *Under Assumption 5 (i), for the finite-difference based algorithm specified by* (3.10)*,*

$$\mathbb{E}\,T_{\tau(\epsilon)} = O(d^{2p+5-\frac{3}{2}q}\epsilon^{-\frac{2(r+2\rho p+\rho(2-q)+1)}{(2\rho)\wedge(\beta+r)}}).$$

Under Assumption 5 (ii), for the finite-difference based algorithm specified by (3.10)*,*

$$\mathbb{E}\,T_{\tau(\epsilon)} = O(d^{5-\frac{3}{2}q}\epsilon^{-\frac{2(r+\rho(2-q)+1)}{(2\rho)\wedge(\beta+r)}}\exp((m_0 d^2+1)\kappa^{\frac{2\rho}{(2\rho)\wedge(\beta+r)}}\epsilon^{-\frac{4\rho}{(2\rho)\wedge(\beta+r)}}\log\alpha)).$$

We are now in a position to identify an optimal set of algorithm parameters such that the order of expected computation cost is minimized. According to Theorem 8, under Assumption 5 (i), this problem is equivalent to the following minimization problem:

$$\begin{aligned} &\min_{r,\rho,\beta} \frac{r+\rho(2p+2-q)+1}{(2\rho)\wedge(\beta+r)} \\ &s.t.\ r \geq 0, \rho > 0, \frac{1}{2} < \beta \leq 1. \end{aligned} \tag{3.15}$$

Proposition 9. *The optimal objective value of the problem (3.15) is* $p-\frac{q}{2}+2$*, and is achieved when the parameters* r, ρ, β *satisfy*

$$\beta = 1,\ 2\rho = r+1. \tag{3.16}$$

We formally present the algorithm in Algorithm 1 with the optimal choices of parameters.

Algorithm 1 Simulation-optimization algorithm with finite difference gradient estimator

```
Input: Number of iterations $N$, initial point $\theta_0$, parameters $m_0, N_0 \in (0,\infty)$, $r \geq 0$, $\rho = \frac{1+r}{2}$, $\gamma_0 \in (\frac{1+r}{\mu}, \infty)$
Output: $\theta_N$
for $t = 1$ to $N-1$ do
    Set $m_t = \lceil m_0 d^2 t^{2\rho} \rceil$, and $N_t = \left\lceil N_0 d^{5-\frac{3}{2}q} t^{r+\rho(2-q)} \right\rceil$
    Generate a sequence $\{Y_{m_t,l}\}_{l=1}^{N_t}$ of independent copies of $Y_{m_t}$ and sample independent vectors $\{Z_{t,l}\}_{l=1}^{N_t}$ from $\nu$.
    Update $\theta_t$ by $\theta_{t+1} = \mathrm{pr}_\Theta(\theta_t - \gamma_t H_t(\theta_t))$, where
        $H_t(\theta) = \frac{1}{N_t} \sum_{l=1}^{N_t} \frac{G_{m_t}(\theta + h_t Z_{t,l}, Y_{m_t,l}) - G_{m_t}(\theta, Y_{m_t,l})}{h_t} Z_{t,l}$
end for
return $\theta_N$
```

Now we introduce the asymptotic distribution of θ_t for Algorithm 1. First we study the asymptotic distribution of θ_t as t goes to infinity. Then we derive a central limit theorem for the best estimator available with the given budget C going to infinity.

Theorem 10. *Suppose that all conditions of Theorem 4 hold. Denote* H^* *as the Hessian matrix for* $g(\theta)$ *at* θ^* *and* $\tilde{H} := H^* - \frac{1+r}{2\gamma_0} I$*. Denote* $C' := \lim_{n\to\infty} n^{1/2} \left(\nabla_\theta g_n(\theta^*) - \nabla_\theta g(\theta^*)\right)$*. Under suitable regularity assumptions, for the finite difference regime in Theorem 4 with* $\beta = 1$ *and* $2\rho = r+1$*,*

$$n^{(1+r)/2}(\theta_n - \theta^*) \xrightarrow{d} N(-\tilde{H}^{-1}(\frac{1}{2}\mathbb{E}[ZZ^\top H^* Z] + m_0^{-\frac{1}{2}} d^{-1} C'), \Sigma) \tag{3.17}$$

where

$$\Sigma = \frac{\gamma_0}{N_0}\int_0^\infty \exp(-\tilde{H}u)\Omega \exp\left(-\tilde{H}^\top u\right) du$$

and

$$\Omega = \mathbb{E}[(Z^\top \nabla_\theta G(\theta^*, Y))^2 ZZ^\top].$$

Especially, if $Z \sim N(0, I_{d\times d})$*, we have*

$$n^{(1+r)/2}(\theta_n - \theta^*) \xrightarrow{d} N(-m_0^{-\frac{1}{2}} d^{-1}\tilde{H}^{-1}C', \Sigma)$$

and

$$\Omega_{ij} = \begin{cases} 3\,\mathbb{E}(\frac{\partial}{\partial\theta_i}G(\theta^*, Y))^2 + \sum_{k\neq i}\mathbb{E}(\frac{\partial}{\partial\theta_k}G(\theta^*, Y))^2, & \text{if } i = j \\ 2\,\mathbb{E}\,\frac{\partial}{\partial\theta_i}G(\theta^*, Y)\frac{\partial}{\partial\theta_j}G(\theta^*, Y), & \text{if } i \neq j. \end{cases}$$

We now change our lens to the available computation budget C and derive a central limit theorem for the best estimator available with the given budget.

Theorem 11. *Suppose that all conditions of Theorem 10 are satisfied. Let* C *be the computation budget and* $n(C) := \sup\{n \geq 1 : 2\sum_{j=1}^n N_j C_{m_j} \leq C\}$*. If* $n^{-p}\,\mathbb{E}[C_n] \to \kappa_1$ *for some* $p > 0$ *and* $\kappa_1 > 0$*, then*

$$(\frac{C}{2\kappa_1 N_0 d^{2(p+1)} m_0^p (p+1)})^{1/2(p+1)}(\theta_{n(C)} - \theta^*) \xrightarrow{d} N(-\tilde{H}^{-1}(\frac{1}{2}\,\mathbb{E}[ZZ^\top H^* Z] + m_0^{-\frac{1}{2}} d^{-1}C'), \Sigma)$$

where

$$\Sigma = \frac{\gamma_0}{N_0}\int_0^\infty \exp(-\tilde{H}u)\Omega \exp\left(-\tilde{H}^\top u\right) du$$

and

$$\Omega = \mathbb{E}[(Z^\top \nabla_\theta G(\theta^*, Y))^2 ZZ^\top].$$

See appendices for the detailed assumptions and proofs of Theorem 10 and 11.

3.4 Simulation-Optimization Algorithms with Multilevel Gradient Estimators

In Section 3.3, in the gradient-based algorithms, the gradient estimator used in each step is constructed by a single approximating system. In this section, we construct multilevel finite difference gradient estimators for which each gradient estimator utilizes multiple approximating systems at different resolutions. We then propose a simulation-optimization algorithm using multilevel gradient estimators. We show that the use of multilevel gradient estimators can improve the performance of the simulation-optimization algorithms compared with the results in Section 3.3. The improved performance is demonstrated by a faster order of convergence rate.

In our basic framework, $G(\theta, Y)$ is approximated by the sequence $(G_k(\theta, Y_k) : k \geq 0)$. The approximation can either be applied to the simulation logic G or the random object Y, or both. In this section, for the convenience of notation in constructing multilevel gradient estimators, we drop the dependence of Y_k on k and summarize all the dependence of the k-th approximating system on k into the function G_k. That is, we rewrite $\tilde{G}_k(\theta, Y) := G_k(\theta, Y_k)$, in which Y is a (possibly infinite-dimensional) random object that is shared by all approximating systems. In this new notation, we note that the k-th system does not necessarily need to use all the information in Y.

Construction of Multilevel FD Gradient Estimator, Assumptions and Example

We propose an algorithm that uses multilevel FD gradient estimators. Consider an increasing positive integer sequence $(m_t : t \geq 1)$ and a non-increasing sequence of positive smoothing parameters $(h_t : t \geq 1)$. At the t-th step, the algorithm updates the θ_{t-1} from the previous step by using the information of function values from the approximating systems $1, \ldots, m_t$. Specifically, the algorithm generates $N_{t,k}$ independent copies of Y for $k = 1, ...m_t$, denoted as $\{\{Y_{t,k,l}\}_{l=1}^{N_{t,k}}\}_{k=1}^{m_t}$ and sample independent vectors $\{\{Z_{t,k,l}\}_{l=1}^{N_{t,k}}\}_{k=1}^{m_t}$ from ν. As introduced in Section 3.3, this distribution ν represents the distribution of the random direction drawn to construct FD gradient estimators. We obtain the gradient estimator

$$H_t(\theta) = \sum_{k=1}^{m_t} \frac{1}{N_{t,k}} \sum_{l=1}^{N_{t,k}} (F_k^{\text{sm}}(\theta; h_t, Z_{t,k,l}, Y_{t,k,l}) - F_{k-1}^{\text{sm}}(\theta; h_t, Z_{t,k,l}, Y_{t,k,l}))$$

where

$$F_k^{\text{sm}}(\theta; h_t, Z_{t,k,l}, Y_{t,k,l}) = \frac{G_k(\theta + h_t Z_{t,k,l}, Y_{t,k,l}) - G_k(\theta, Y_{t,k,l})}{h_t} Z_{t,k,l}$$

and updates θ_t as

$$\theta_{t+1} = \text{pr}_\Theta(\theta_t - \gamma_t H_t(\theta_t)).$$

The gradient estimator $H_t(\theta)$ is the *multilevel gradient estimator* used in the t-th step. Intuitively, the multilevel construction couples the adjacent systems G_k and G_{k-1} to achieve a better bias-variance trade-off for the gradient estimator.

We hope to provide answers to the following questions for the gradient-based algorithm utilizing multilevel FD gradient estimators:

- What are the optimal choices of parameter m_t, $N_{t,k}$ and γ_t at the t-th step?
- What is the convergence rate of θ_t to θ^*?
- Compared with simulation-optimization algorithm we discussed in Section 3.3, does multilevel method based on FD estimator expect to save cumulative computation cost that is needed to achieve a given precision level?

- How does the cumulative computation cost depend on the dimension d?

Before proceeding into the algorithm analysis, we make the following assumption on the approximation $G_k(\cdot, Y)$, which quantifies not only the bias of $G_k(\theta, Y)$, but also L^2-distance between expected performance function between $G_k(\theta, Y)$ and $G(\theta, Y)$:

Assumption 6. *There exist constants $M > 1$ and $\alpha \geq \eta > 0$ such that for all $k \in \mathbb{N}$ and $\theta \in \Theta$*

(i) $|\mathbb{E}\left[G_k(\theta, Y) - G(\theta, Y)\right]| \leq M^{-k\alpha}$

(ii) $\mathbb{E}\left[|G_k(\theta, Y) - G(\theta, Y) - \mathbb{E}[G_k(\theta, Y) - G(\theta, Y)]\,|^2\right]^{1/2} \leq M^{-k\eta}$.

We illustrate this assumption by considering a simulation-optimization problem in which the original objective function g depends on the sample path of the solution of a general Stochastic Differential Equation (SDE). The solution of an SDE is an infinite-dimensional random object that in general cannot be exactly simulated in finite time. (Exact and practically implementable simulation algorithms are available, but only for a subset of SDE models.) A standard practice in Monte Carlo simulation is to discretize the SDE, which serves as an approximation for the original SDE. In this example, we consider the approximating systems to use different resolutions of discretizations of SDE. We consider the Euler-Maruyama discretization scheme of a diffusion process $X = (X_t : t \geq 0)$, which is the solution of an SDE. We first introduce some notation, namely for $x \in \mathbb{R}^q$,

$$f(x) = \begin{pmatrix} b_1(x) & \sigma_{11}(x) & \cdots & \sigma_{1q'}(x) \\ b_2(x) & \sigma_{21}(x) & \cdots & \sigma_{2q'}(x) \\ \vdots & \vdots & \cdots & \vdots \\ b_q(x) & \sigma_{q1}(x) & \cdots & \sigma_{qq'}(x) \end{pmatrix}$$

and $dY_t = \left(dt, dW_t^1, \ldots, dW_t^{q'}\right)^T$ where $b : \mathbb{R}^q \to \mathbb{R}^q, \sigma : \mathbb{R}^q \to \mathbb{R}^q \times \mathbb{R}^{q'}$. The dynamic of X will be written in the compact form

$$\forall t \in [0, T], \quad X_t = x + \int_0^t f\left(X_s\right) dY_s.$$

The Euler-Maruyama scheme with time step $\Delta_n = T/M^n, t_i = i\Delta_n, i = 0, \ldots .M^n$, $\phi_n(s) = \sup\{t_i : t_i \leq s\}$ is given by

$$X_t^n = x + \int_0^t f\left(X^n_{\phi_n(s)}(s)\right) dY_s$$

Under some smoothness assumptions on the coefficients $b(x)$ and $\sigma(x)$, according to [58], there exists $c \in (0, \infty)$ such that for every $k \in \mathbb{N}$,

$$\mathbb{E}[\|X_T^k - X_T\|^2]^{\frac{1}{2}} \leq \frac{c}{M^{\frac{k}{2}}}.$$

The system performance $V : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}$ maps the decision variable $\theta \in \mathbb{R}^d$ and the realization of the random vector X_T into a real-valued performance. Since X_T does not have a closed-form representation, we use X_T^k to evaluate X_T and use $G_k(\theta, X_T) := V(\theta, X_T^k)$ to approximate $G(\theta, X_T)$. We assume that there exists a constant $L > 0$, such that for every $\theta \in \Theta$ and every $x, y \in \mathbb{R}^q$, $|V(\theta, x) - V(\theta, y)| \le L\|x - y\|$. Therefore, we have

$$|\,\mathbb{E}\,V(\theta, X_T) - \mathbb{E}\,V(\theta, X_T^k)| \le L\,\mathbb{E}\,\|X_T - X_T^k\| \le LcM^{-\frac{k}{2}},$$

and

$$\mathbb{E}[|V(\theta, X_T) - V(\theta, X_T^k)|^2]^{\frac{1}{2}} \le L\,\mathbb{E}[\|X_T - X_T^k\|^2]^{\frac{1}{2}} \le LcM^{-\frac{k}{2}}.$$

Therefore, the above analysis suggests that Assumption 6 stands with $\alpha = \frac{1}{2}$ and $\eta = \frac{1}{2}$.

Algorithm Convergence and Cost

We now state a convergence rate result on the algorithm with multilevel FD gradient estimator and show that the algorithm will eventually recover θ^* with properly chosen algorithm parameters. The convergence rate of θ_t relies on the choices of m_t, h_t, γ_t and $N_{t,k}$ at each step. For a set of parameters $\gamma_0, N_0 > 0$, $\beta, r, \rho \ge 0$, we consider algorithm parameters given by

$$m_t = \lceil \frac{2}{\alpha} \log_M(dt^\rho) \rceil, \quad h_t = d^{-\frac{3}{2}} t^{-\rho}, \quad \gamma_t = \gamma_0 \frac{1}{t^\beta}, \quad N_{t,k} = \lceil \kappa_t M^{-k(\eta+1/2)} \rceil, \tag{3.18}$$

where

$$\kappa_t = \begin{cases} d^4 t^{r+2\rho} M^{m_t\left(\frac{1}{2}-\eta\right)_+}, & \text{if } \eta \ne \frac{1}{2} \\ d^4 t^{r+2\rho} m_t, & \text{if } \eta = \frac{1}{2}. \end{cases}$$

In (3.18), $N_{t,k}$ increases as time step t increases, fixing the level k. The reason why $N_{t,k}$ increases as t increases is that we want to control the variance of our gradient estimator at each time step in the optimization algorithm based on multilevel FD estimator. On the other hand, $N_{t,k}$ will decrease as k, the level of approximation, increases. Given the parameters, we formally present the multilevel FD algorithm.

Theorem 12. *Suppose that $\beta \in (\frac{1}{2}, 1]$, $r \ge 0$, $\rho, \gamma_0, N_0 \in (0, \infty)$. If $\beta = 1$, suppose additionally that $\gamma_0 \in (\max\{\frac{2\rho}{\mu}, \frac{1+r}{\mu}\}, \infty)$. Under Assumption 1, 4 and 6, there exists a $\kappa \in (0, \infty)$ independent of d such that for all $t \in \mathbb{N}$,*

$$\mathbb{E}\left[g(\theta_t) - g(\theta^*)\right] \le \frac{1}{2} L\kappa\, t^{-(2\rho)\wedge(\beta+r)},$$

and

$$\mathbb{E}\left[\|\theta_t - \theta^*\|^2\right] \le \kappa t^{-(2\rho)\wedge(\beta+r)}.$$

Algorithm 2 Simulation optimization with multilevel finite difference estimator

```
Input:Number of iterations $N$, initial point $\theta_0$, parameters $N_0 \in (0, \infty)$, $r \geq 0$, $\rho = \frac{1+r}{2}$, $\gamma_0 \in (\frac{1+r}{\mu}, \infty)$
Output: $\theta_N$
for $t = 1$ to $N - 1$ do
    Set $m_t = \lceil \frac{2}{\alpha} \log_M(dt^\rho) \rceil$, $h_t = d^{-\frac{3}{2}} t^{-\rho}$ and $N_{t,k} = \kappa_t M^{-k(\eta+1/2)}$, where
        $$\kappa_t = \begin{cases} d^4 t^{r+2\rho} M^{m_t\left(\frac{1}{2}-\eta\right)^+}, & \text{if } \eta \neq \frac{1}{2} \\ d^4 t^{r+2\rho} m_t, & \text{if } \eta = \frac{1}{2} \end{cases}$$
    generates $N_{t,k}$ independent copies of $Y$ for $k = 1, ...m_t$, denoted by $\{(Y_{t,k,l})_{l=1}^{N_{t,k}}\}_{k=1}^{m_t}$
    Update $\theta_{t+1} = \mathrm{pr}_\Theta(\theta_t - \frac{\gamma_0}{t} H_t(\theta_t))$, where
        $$H_t(\theta) = \sum_{k=1}^{m_t} \frac{1}{N_{t,k}} \sum_{\ell=1}^{N_{t,k}} (F_k^{\mathrm{sm}}(\theta; h_t, Z_{t,k,l}, Y_{t,k,l}) - F_{k-1}^{\mathrm{sm}}(\theta; h_t, Z_{t,k,l}, Y_{t,k,l}))$$
end for
return $\theta_N$
```

According to Theorem 4 and Theorem 12, the algorithm with standard FD estimator in Section 3.3 and the algorithm with multilevel FD estimator in Section 3.4 have the same order of MSE convergence rate $t^{-(2\rho)\wedge(\beta+r)}$ with respect to the time step t. However, the MSE convergence rate with respect to computational budget C for the two algorithms are different. Equivalently speaking, given the same precision level ϵ, the number of iteration steps for two algorithms are the same, but the computational cost required are different. In what follows, we present the results on the expected computational cost required to achieve a certain precision level for the algorithm with the use of multilevel FD estimator. We compare the expected computational cost for the two algorithms.

Recall that C_k is defined as the computational cost for generating a single simulation run of $G_k(\theta, Y)$. This computational cost C_k is allowed to be a random variable itself and we make the following assumptions that characterize the growth rate of the expected cost when $k \to \infty$. As an illustration, this assumption holds for the SDE discretization setting.

Assumption 7. *As $k \to \infty$, the sequence of cost $\{C_k : k \geq 1\}$ satisfies $M^{-k}\,\mathbb{E}[C_k] \to \kappa_1$ for some $M > 1$ and $\kappa_1 > 0$.*

In Section 3.3, we assume that the difference between k-th system and the original system has a scale of Ck^{-1} as shown in Assumption 2, and the computational cost for simulating a single copy of the k-th system C_k satisfies $k^{-p}\,\mathbb{E}[C_k] \to \kappa_1 > 0$ as shown in Assumption 5 (i). In this section, we assume that $|\mathbb{E}\left[G_k(\theta, Y) - G(\theta, Y)\right]| \leq M^{-k\alpha}$ and $M^{-k}\,\mathbb{E}[C_k] \to \kappa_1 > 0$. To compare the expected computational cost of two algorithms, we need to rescale the approximating systems and let $p = \frac{1}{\alpha}$. The purpose is to make sure $\mathbb{E}\, C_k$ increases at the same speed with respect to the bias of $G_k(\theta, Y)$ for the problem setting in Section 3.3 and in Section 3.4.

Similar to the definition in (3.13), the notion $\tau(\epsilon)$ is the number of iterations the algorithm takes for θ_t to satisfy a precision level ϵ. The notion $T_{\tau(\epsilon)}$ is the cumulative computation cost for the algorithm by the $\tau(\epsilon)$-th iteration.

Using the lens of computation cost, we provide the following result to quantify $T_{\tau(\epsilon)}$.

Theorem 13. *Set $\beta = 1$ and $2\rho = r + 1$ in Theorem 12,*

$$\mathbb{E}\, T_{\tau(\epsilon)} = \begin{cases} O\left(d^{\frac{2}{\alpha}}\epsilon^{-\frac{2}{\alpha}-\frac{1}{\rho}} + d^{4+\frac{2(1-2\eta)_+}{\alpha}}\epsilon^{-4-\frac{2(1-2\eta)_+}{\alpha}}\right), & \text{if } \eta \neq 1/2 \\ O\left(d^{\frac{2}{\alpha}}\epsilon^{-\frac{2}{\alpha}-\frac{1}{\rho}} + d^4\epsilon^{-4}\log_M(\epsilon^{-1})^2\right), & \text{if } \eta = 1/2. \end{cases}$$

If additionally $\alpha > \frac{1}{2}$ and $\rho > \frac{\alpha}{4\alpha+2(1-2\eta)_+-2}$,

$$\mathbb{E}\, T_{\tau(\epsilon)} = \begin{cases} O(d^4\epsilon^{-4}), & \text{if } \eta > \frac{1}{2}, \\ O(d^4\epsilon^{-4}(\ln(\epsilon^{-1}))^2), & \text{if } \eta = \frac{1}{2}, \\ O(d^{4+\frac{2(1-2\eta)_+}{\alpha}}\epsilon^{-\left(4+\frac{2-4\eta}{\alpha}\right)}), & \text{if } \eta < \frac{1}{2}. \end{cases}$$

Remark 14. *Under Assumption 7, and assuming that ρ is fixed, $r = 2\rho - 1$ and $\beta = 1$ is the optimal choice of algorithm parameters to minimize cumulative computational cost. We give the justification for this claim in the appendices.*

If the multilevel approach is not employed, under the exact same problem setting, the optimal algorithm in Section 3.3 has the cumulative computational cost at the order of $O(d^{\frac{2}{\alpha}+5-\frac{3}{2}q}\epsilon^{-\frac{2}{\alpha}+q-4})$. The key insight of Theorem 13 is that the use of multilevel FD gradient estimators, compared to the use of standard FD gradient estimators under the discontinuity case ($q = 0$), improves the computation cost by a factor of $d\epsilon^{-\frac{2}{\alpha}}$ if $\eta \geq \frac{1}{2}$, or a factor of $d\epsilon^{-\frac{4\eta}{\alpha}}$ better if $\eta < \frac{1}{2}$.

3.5 Numerical Experiments

In this section, we implement the two algorithms constructed in Section 3.3 (with FD gradient estimators) and Section 3.4 (with multilevel FD gradient estimators) on a numerical experiment. We find that the experiment results support our theoretical findings. We study a stylized simulation-optimization problem on portfolio selection, where the underlying prices of assets are driven by stochastic differential equations. We implement the two algorithms and numerically demonstrate the convergence rates. The experiment also shows that the additional use of multilevel gradient estimators can reduce the order of computation cost required to achieve a certain level of precision. The details of implementation of simulation-optimization algorithms and the experiment are posted to [107].

We consider a setting that has an exact closed-form solution so that we can accurately compute the error of the algorithm. Specifically, we consider a portfolio of $d \in \mathbb{N}$ assets whose price process $S(t) = (S_1(t), S_2(t), \dots, S_d(t))$ follows a d-dimensional geometric Brownian motion:

$$dS(t) = \operatorname{diag}(S(t))\mu\, dt + \operatorname{diag}(S(t))V\, dW(t) \tag{3.19}$$

where μ is a d-dimensional vector, V is a $d \times d$ matrix, $W(t)$ is a d-dimensional standard Brownian motion and $\operatorname{diag}(S(t))$ denotes the diagonal matrix in $\mathbb{R}^{d\times d}$ with entries $S_1(t), S_2(t), \dots, S_d(t)$. Besides, we denote $S_f(t)$ as the value of risk-free asset with initial value $S_f(0)$ and risk-free return rate r_f. That is, $S_f(t) = S_f(0)\exp(r_f t)$ for all $t \geq 0$.

The portfolio optimization problem is to minimize

$$\mathbb{E}\Big(\sum_{i=1}^{d} w_i S_i(T) + (1 - \sum_{i=1}^{d} w_i) S_f(T) - \gamma\Big)^2 \tag{3.20}$$

such that

$$\sum_{i=1}^{d} w_i \leq 1, \tag{3.21}$$

and

$$w_i \geq 0,\ i = 1, 2, \dots, d. \tag{3.22}$$

In (3.20), γ represents a given target value of the portfolio at time $t = T$ and w_i is the weight for asset i respectively for the portfolio. In this optimization problem, the decision variable is to select the portfolio weights for the assets at the beginning $t = 0$. The goal is to have the portfolio value achieve a target value γ at time T, while minimizing the variability or risk, represented by the variance of the portfolio value at time T. We require that it is not possible to do a short sale for the assets, so the weight w_i, $i = 1, 2, \dots, d$ must be in the interval $[0, 1]$, as illustrated by (3.21) and (3.22).

We note that this example setting may represent a wide class of simulation-optimization problems that involve stochastic differential equations as infinite-dimensional random objects. In general, such problems do not adopt closed-form solutions, and discretization simulation schemes are often employed to evaluate and approximate the objective function. However, the example in consideration is simple enough, so that exact solutions are available, which facilitate the demonstration of algorithm performance.

We describe how our proposed simulation-optimization algorithms are implemented. We set $w = (w_1, w_2, \dots, w_d)^\top$, $G(w, S(T)) := (\sum_{i=1}^{d} w_i S_i(T) + (1 - \sum_{i=1}^{d} w_i) S_f(T) - \gamma)^2$ and $g(w) := \mathbb{E}\, G(w, S(T))$. The random objective function $G(w, S(T))$ is approximated by the Euler-Maruyama discretization scheme at different resolutions. Based on (3.19), the Euler-Maruyama scheme with time step $\Delta_k = T/M^k, t_j = j\Delta_k, j = 0, \dots M^k$, $\phi_k(s) = \sup\{t_j :$

$t_j \leq s\}$ is given by

$$
\begin{aligned}
S_i^k(t) =& S_i(0) + \mu_i \int_0^t S_i^k(\phi_k(s))ds \\
&+ \sum_{l=1}^{d} V_{i,l} \int_0^t S_l^k(\phi_k(s))dW_j(s)
\end{aligned}
$$

for $i = 1, 2, \ldots, d$. We define

$$
G_k(w, S(T)) := (\sum_{i=1}^{d} w_i S_i^k(T) + (1 - \sum_{i=1}^{d} w_i) S_f(T) - \gamma)^2
$$

as the k-approximating system.

For this simulation optimization problem, $g(\cdot)$ satisfies Assumption 1. Besides, since $S(t)$ follows geometric Brownian motion, according to our analysis in Section 3.4, $G(\cdot, S(T))$ and $\{G_k(\cdot, S(T))\}_{k\in\mathbb{N}}$ satisfy Assumption 6 and 7 with $\alpha = \eta = \frac{1}{2}$. By rescaling the approximating systems, this example satisfies Assumption 2, Assumption 3 with $q = 0$ and Assumption 5 (i) with $p = \frac{1}{\alpha} = 2$. Therefore, simulation optimization algorithm based on regular FD estimator (Algorithm 1) and based on multilevel FD estimator (Algorithm 2) can be applied to this problem with a theoretical guarantee of convergence rate.

Numerical Results with Dimension $d = 2$

First we consider the case when $d = 2$ and implement the simulation-optimization algorithm based on FD gradient estimator, provided by Algorithm 1. We set the initial parameters of this algorithm to be

$$
N_0 = 5, r = 0, \rho = \frac{1}{2}, \gamma_0 = 10.
$$

The random direction Z is chosen to be uniformly distributed on the l_2-sphere of radius $\sqrt{d}$. Besides we set the initial solution $\theta_0 = (0, 0)$ and denote $\theta_t = (w_{1,t}, w_{2,t})$ as the solution after t iterations by Algorithm 1. According to Theorem 4, for θ_t, we are expected to see that

$$
\mathbb{E}\left[\|\theta_t - w^*\|^2\right] \leq \kappa t^{-1}, \tag{3.23}
$$

where w^* represents the optimal solution of the optimization problem (3.20)-(3.22). Figure 3.1 shows the log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ for $t = 1, \ldots, 200$ steps. To approximate $\mathbb{E}\,\|\theta_t - w^*\|^2$, we use $N = 200$ independent simulation replications, i.e., we implement $N = 200$ independent solving processes. A red straight line with slope -1 is also plotted in Figure 3.1, to demonstrate the convergence rate of θ_t. From Figure 3.1, when t is small, the decreasing rate of the mean square error of θ_t is slower than t^{-1}. When t becomes larger, the mean square error of θ_t demonstrates a decreasing rate at an order that becomes closer to $1/t$, which tends to match the theoretical result in Theorem 4.

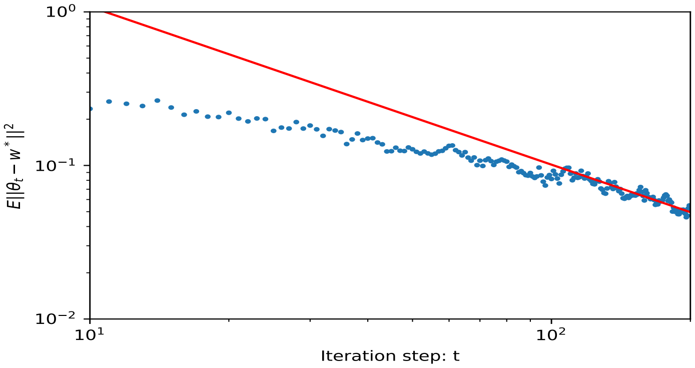


Figure 3.1: The log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ for $t = 1, \ldots, 200$ steps by Algorithm 1 when $d = 2$. $\mathbb{E}\,\|\theta_t - w^*\|^2$ is approximated by 200 independent simulation replications. The y-value of the scatters represent the estimated value of $\mathbb{E}\,\|\theta_t - w^*\|^2$ at the iteration step t. The red straight line has slope -1. The empirical trajectory demonstrates that the decreasing rate of $\mathbb{E}\,\|\theta_t - w^*\|^2$ as t increases is closer and closer to $1/t$.

In order to show that the central limit theorem in Theorem 10 actually stands, we plot in Figure 3.2 the histogram of $t^{\frac{1}{2}}(w_{1t} - w_1^*)$ and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$. For the purpose of comparison, we plot in Figure 3.2 using red dashed curve for a normal distribution density with mean and variance matching that of the plotted histogram of $t^{\frac{1}{2}}(w_{1t} - w_1^*)$ and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$. We also plot the smoothed version of the estimated density function for $t^{\frac{1}{2}}(w_{1t} - w_1^*)$ and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$ using blue curve for illustration. Figure 3.2 suggests that the distribution of $t^{\frac{1}{2}}(w_{1t} - w_1^*)$ and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$ are close to normal distribution, which tends to support our proved central limit theorem results.

Numerical Results with Dimension $d = 20$

We consider the case when $d = 20$ and present numerical results. The numerical results include three parts. In the first part, we compare the convergence rate with respect to time step t for Algorithm 1 and Algorithm 2 when $d = 20$. Second, we compare the performance of Algorithm 1 and Algorithm 2 when the same computational budget C is provided to each algorithm. In the third part, we compare the convergence rate with respect to computational budget C for Algorithm 2 for $d = 5$ and $d = 20$ case and show the influence of dimension of decision variable on Algorithm 2's performance.

We implement Algorithm 1 under the case $d = 20$. The initial parameters of this algorithm is set to be the same as when we implement $d = 2$ case. Figure 3.3a shows the log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ for $t = 1, \ldots, 200$ steps when $d = 20$ for Algorithm 1. A red line of slope -1 is plotted for comparison. From Figure 3.3a, as $t \to \infty$, $\mathbb{E}\,\|\theta_t - w^*\|^2$ demonstrates a decreasing rate at the order of roughly $1/t$, matching our theory results.

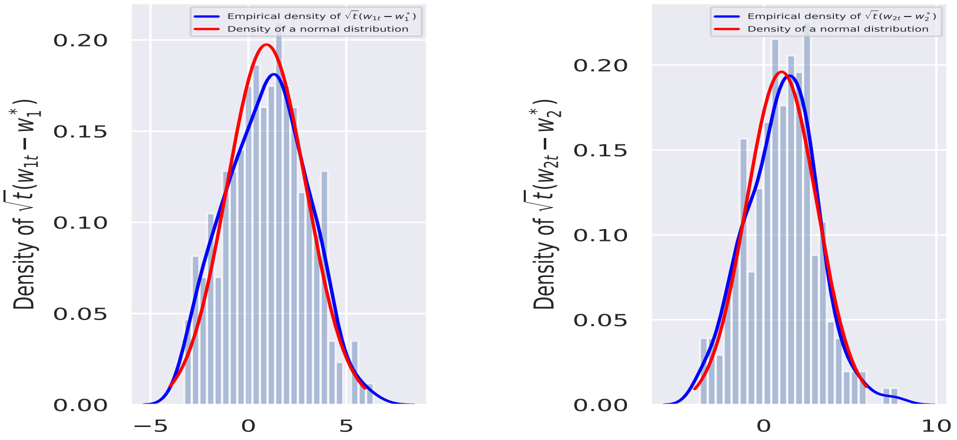


Figure 3.2: Rescaled histogram of $t^{\frac{1}{2}}(w_{1t} - w_1^*)$ on the left and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$ on the right, with $t = 200$, constructed by $N = 200$ independent solving processes. Red curves (that are symmetric) represent for the normal distribution density function with mean and variance estimated from data. Blue curves represent for the estimated density function of $t^{\frac{1}{2}}(w_{1t}-w_1^*)$ and $t^{\frac{1}{2}}(w_{2t} - w_2^*)$ by using kernel density estimation.

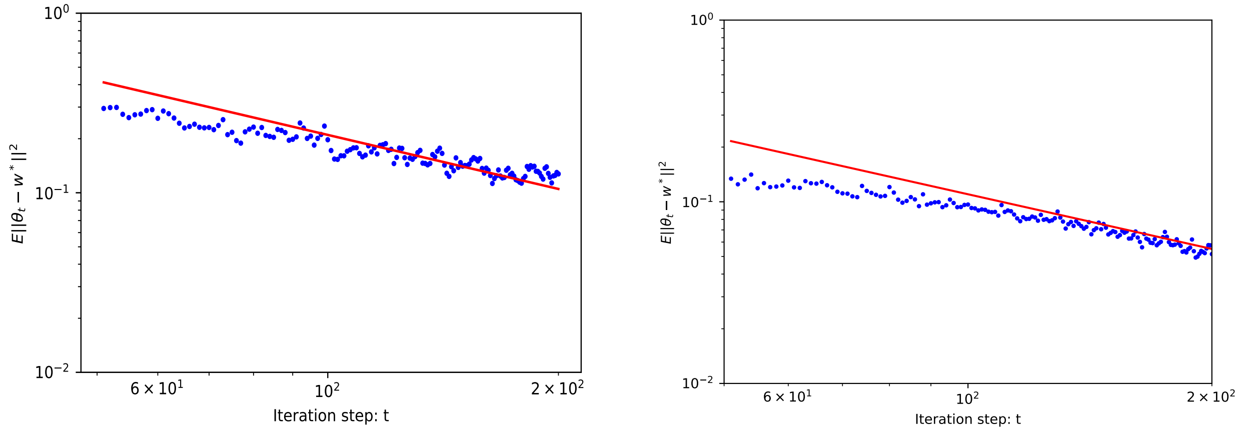
Next we turn our attention to the simulation-optimization algorithm based on multilevel finite difference (FD) gradient estimator, provided by Algorithm 2. We set the initial parameters of this algorithm to be

$$N_0 = 2, r = 0, \rho = \frac{1}{2}, \gamma_0 = 10.$$

We keep the same r, ρ and γ_0 as in the experiment for regular FD estimator but N_0 has changed. Same with the numerical experiments based on Algorithm 1 with $d = 20$, we set the initial point $\theta_0 = (0, 0, ...0)^\top$. In Figure 3.3b, we draw the log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ for $t = 1, \ldots, 200$ steps when $d = 20$ for Algorithm 2. A red line of slope -1 is plotted for comparison. Figure 3.3b illustrates that as $t \to \infty$, $\mathbb{E}\,\|\theta_t - w^*\|^2$ has a decreasing rate at the order of roughly $1/t$, matching our theory results.

Next, we compare the performance of the optimization algorithm based on multilevel FD gradient estimator (Algorithm 2) and the optimization algorithm based on regular FD gradient estimator (Algorithm 1) with the same dimension, when the same computational budget is provided to each algorithm.

Define $n_{ml}(C)$ as the number of iteration steps achieved by Algorithm 2 when C computational budget is used. Then denote $\theta^{ml}_{n_{ml}(C)}$ as the estimator provided by Algorithm 2 when C computational budget is used. Meanwhile, define $n(C)$ as the number of iteration steps achieved by Algorithm 1 when C computational budget is used. Denote $\theta_{n(C)}$ as the estimator provided by Algorithm 1 when C computational budget is used. In Figure 3.4, we draw a scatter plot to demonstrate the mean square error of $\theta^{ml}_{n_{ml}(C)}$ and $\theta_{n(C)}$ respectively



(a) FD estimator (Algorithm 1)

(b) Multilevel FD estimator (Algorithm 2)

Figure 3.3: The log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ for $t = 1, \ldots, 200$ steps by Algorithm 1 and 2 respectively. The dimension d is set to be 20. $\mathbb{E}\,\|\theta_t - w^*\|^2$ is approximated by 200 independent simulation replications. The y-value of the scatters represent the estimated value of $\mathbb{E}\,\|\theta_t - w^*\|^2$ at the iteration step t. The red straight line has slope -1. Figure 3.3a is log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ by Algorithm 1 when $d = 20$. Figure 3.3b is log-log plot of the empirical trajectory of $\mathbb{E}\,\|\theta_t - w^*\|^2$ by Algorithm 2 when $d = 20$. The empirical trajectories demonstrate that the decreasing rate of $\mathbb{E}\,\|\theta_t - w^*\|^2$ as t increases is closer and closer to $1/t$, for both Algorithm 1 and Algorithm 2.

as a function of C. The scatter plot is done at a log-log scale. The mean square errors are computed from 200 independent solving processes. Shown by Figure 3.4, as the computational budget C increases, the decreasing slope of the mean square error of $\theta^{ml}_{n_{ml}(C)}$ is close to $-1/3$, while the decreasing slope of the mean square error of $\theta_{n(C)}$ is close to $-1/4$. The results match the theory provided in Theorem 8 and Theorem 13, showing that the use of multilevel gradient estimator improves the order of computational efficiency.

For simulation-optimization algorithm based on multilevel FD estimator (Algorithm 2), the computation cost to achieve a given precision level is proved to have a polynomial order of dependence on the dimension of decision variable d. To show the influence of dimension of decision variable on Algorithm 2's performance, we implement Algorithm 2 for $d = 5$ and $d = 20$. The initial parameters N_0, r, ρ, γ_0 are set to be the same for $d = 5$ and $d = 20$ case. In Figure 3.5, we draw the scatter plot for the mean square error of $\theta^{ml}_{n_{ml}(C)}$ as a function of computational budget C for $d = 5$ and $d = 20$ case respectively. Shown by Figure 3.5, for both $d = 5$ and $d = 20$ case, as the computational budget C increases, the decreasing slope of the mean square error is close to $-1/3$. This matches the theory in Theorem 13. Given the same computational budget C, for $d = 5$ case, the mean square error of $\theta^{ml}_{n_{ml}(C)}$ is approximately 57.08% less than the mean square error in $d = 20$ case.

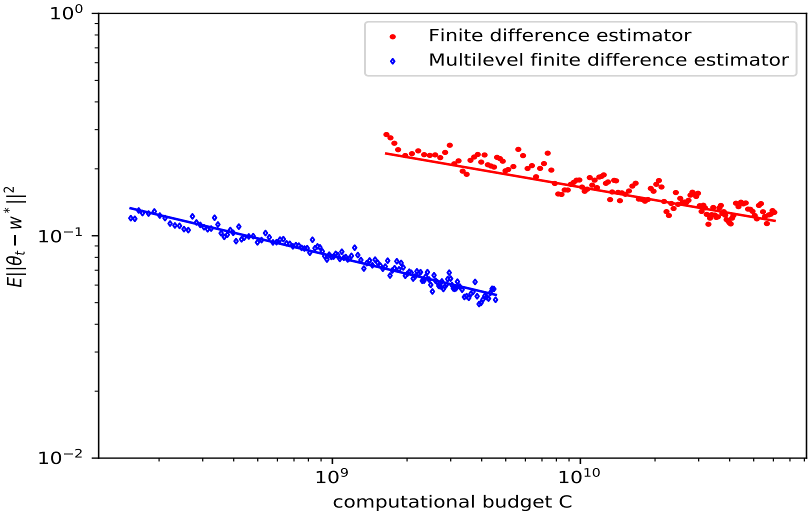


Figure 3.4: Mean square error of $\theta_{n(C)}$ and $\theta^{ml}_{n_{ml}(C)}$ given computational budget C. $\theta^{ml}_{n_{ml}(C)}$ and $\theta_{n(C)}$ are the estimators of w^* achieved by Algorithm 2 and 1 respectively, when C computational budget is used. The dimension d is set to be 20. The mean square errors are computed from 200 independent solving processes. The lower line represents the decreasing slope of the mean square error of $\theta^{ml}_{n_{ml}(C)}$. The upper line represents the decreasing slope of the mean square error of $\theta_{n(C)}$.

3.6 Conclusion

In this chapter, we present a framework to use a sequence of approximating systems to optimize a stochastic system that has complicated stochastic structure and cannot be simulated exactly with finite computational cost. With this framework, we propose new gradient-based simulation-optimization algorithms that utilize the approximations with increasing resolution and higher simulation cost to construct stochastic gradients and perform gradient search. To circumvent the challenge that the objective functions associated with the approximating systems are discontinuous, we use the finite difference method to construct gradient estimators for approximating systems. Under the assumption that the objective function of the original system is strongly convex and smooth, we prove algorithm convergence, convergence rate, and optimality of algorithm design, without any assumption on convexity or continuity with the approximating systems. We demonstrate the dependence on the dimension of the decision variable for the algorithm performance and optimal parameters choices. We then present a multilevel version of the proposed algorithms to further improve convergence rates, when in addition the sequence of approximations can be naturally coupled. We prove theoretically and then show empirically that the additional use of multilevel structure can further improve the computational efficiency of the proposed simulation-optimization algorithms.

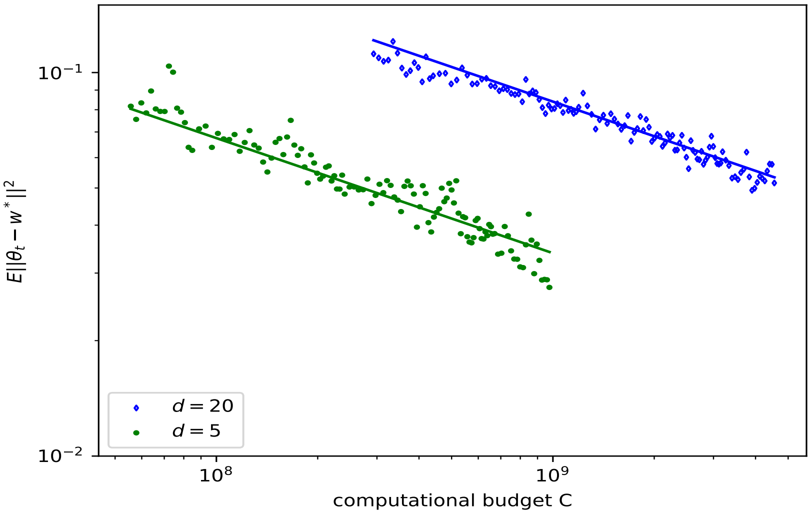


Figure 3.5: Mean square error of the estimator provided by Algorithm 2 given computational budget C. The dimension is set to be 5 and 20 respectively. The mean square errors are computed from 200 independent solving processes. The lower line represents the decreasing slope of the mean square error when $d = 5$. The upper line represents the decreasing slope of the mean square error when $d = 20$.

Appendices

3.A Auxiliary Lemma

Lemma 1. *Suppose that Assumption 1 holds for $\Theta = \mathbb{R}^d$. Consider a sequence $\{\theta_t\}_{t\in\mathbb{N}}$ iteratively defined by*

$$\theta_{t+1} = \theta_t - \gamma_t(\nabla_\theta g(\theta_t) + B_t + V_t). \tag{3.24}$$

Define $\mathcal{F}_t := \mathcal{F}(\theta_0, \theta_1, ...\theta_{t-1}, \theta_t)$. Assume that B_t is $\mathcal{F}_t$ measurable and $\mathbb{E}[V_t|\mathcal{F}_t] = 0$. Furthermore, assume that there exist constants $\beta \in (\frac{1}{2}, 1]$, $r \in \mathbb{R}$, $\rho, \gamma_0, K_1, K_2 \in (0, \infty)$ such that

$$\begin{aligned} \gamma_t &= \gamma_0 t^{-\beta}, \\ \|B_t\| &\leq K_1 t^{-\rho}, \\ \mathbb{E}[\|V_t\|^2|\mathcal{F}_t] &\leq K_2 t^{-r}. \end{aligned} \tag{3.25}$$

We require that $\beta + r > 0$. If $\beta = 1$, require additionally that $\gamma_0 \in (\max\{\frac{2\rho}{\mu}, \frac{1+r}{\mu}\}, \infty)$. Then there exists a $\kappa \in (0, \infty)$ such that for all $t \in \mathbb{N}$,

$$\mathbb{E}\left[\|\theta_t - \theta^*\|^2\right] \leq \kappa t^{-(2\rho)\wedge(\beta+r)}, \tag{3.26}$$

and

$$\mathbb{E}\left[g(\theta_t) - g(\theta^*)\right] \leq \frac{1}{2}L\kappa\, t^{-(2\rho)\wedge(\beta+r)}. \tag{3.27}$$

Proof of Lemma 1.

$$\begin{aligned} &\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2|\theta_t] - \|\theta_t - \theta^*\|^2 \\ =&2(\theta_t - \theta^*)^\top \mathbb{E}[\theta_{t+1} - \theta_t|\theta_t] + \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2|\theta_t] \\ =&-2\gamma_t(\theta_t - \theta^*)^\top \mathbb{E}[\nabla_\theta g(\theta_t) + B_t|\theta_t] + \gamma_t^2\, \mathbb{E}[\|\nabla_\theta g(\theta_t) + B_t + V_t\|^2|\theta_t] \\ \leq&-2\mu\gamma_t\|\theta_t - \theta^*\|^2 - 2\gamma_t(\theta_t - \theta^*)^\top B_t + 3\gamma_t^2(\|\nabla_\theta g(\theta_t)\|^2 + \|B_t\|^2 + \mathbb{E}[|V_t\|^2|\theta_t]) \\ \leq&-2\mu\gamma_t\|\theta_t - \theta^*\|^2 - 2\gamma_t(\theta_t - \theta^*)^\top B_t + 3L^2\gamma_t^2\|\theta_t - \theta^*\|^2 + 3\gamma_t^2(\|B_t\|^2 + \mathbb{E}[\|V_t\|^2|\theta_t]), \end{aligned} \tag{3.28}$$

where we use the fact that $(\theta - \theta^*)^\top \nabla_\theta g(\theta) \geq \mu \|\theta - \theta^*\|^2$ and $\|\nabla_\theta g(\theta)\| \leq L\|\theta - \theta^*\|$. Using the inequality that $2a^\top b \leq (\|a\|^2 + \|b\|^2)$ with $a = \sqrt{\frac{\mu}{2}}(\theta_t - \theta^*)$ and $b = \sqrt{\frac{2}{\mu}} B_t$, we arrive at

$$-2(\theta_t - \theta^*)^\top B_t \leq \frac{\mu}{2}\|\theta_t - \theta^*\|^2 + \frac{2}{\mu}\|B_t\|^2.$$

Combining it with (3.28), we have

$$\begin{aligned}
&\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2 | \theta_t] - \|\theta_t - \theta^*\|^2 \\
\leq& -2\mu\gamma_t\|\theta_t - \theta^*\|^2 + \frac{\mu}{2}\gamma_t\|\theta_t - \theta^*\|^2 + \frac{2\gamma_t}{\mu}\|B_t\|^2 + 3L^2\gamma_t^2\|\theta_t - \theta^*\|^2 + 3\gamma_t^2(\|B_t\|^2 + \mathbb{E}[\|V_t\|^2|\theta_t]) \\
=& -\frac{3}{2}\mu\gamma_t\|\theta_t - \theta^*\|^2 + \frac{2\gamma_t}{\mu}\|B_t\|^2 + 3L^2\gamma_t^2\|\theta_t - \theta^*\|^2 + 3\gamma_t^2(\|B_t\|^2 + \mathbb{E}[\|V_t\|^2|\theta_t]).
\end{aligned}$$

Now consider our assumption that $\gamma_t = \gamma_0 t^{-\beta}$, $\|B_t\| \leq K_1 t^{-\rho}$ and $\mathbb{E}[\|V_t\|^2|\theta_t] \leq K_2 t^{-r}$. Since $\beta > 0$, we can find t_0 such that $3L^2\gamma_0 t_0^{-\beta} \leq \frac{1}{2}\mu$, and $3\gamma_0 t_0^{-\beta} \leq \frac{1}{\mu}$. Therefore, for $t \geq t_0$,

$$\begin{aligned}
&\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2 | \theta_t] - \|\theta_t - \theta^*\|^2 \\
\leq& -\mu\gamma_t\|\theta_t - \theta^*\|^2 + \frac{3\gamma_t}{\mu}\|B_t\|^2 + 3\gamma_t^2\,\mathbb{E}[\|V_t\|^2|\theta_t] \\
\leq& -\mu\gamma_0 t^{-\beta}\|\theta_t - \theta^*\|^2 + \frac{3\gamma_0 K_1^2}{\mu} t^{-2\rho-\beta} + 3K_2\gamma_0^2 t^{-2\beta-r}.
\end{aligned}$$

By taking expectation on both sides and moving $\mathbb{E}\,\|\theta_t - \theta^*\|^2$ to the right side, we get the recursion equation

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] \leq (1 - \mu\gamma_0 t^{-\beta})\,\mathbb{E}[\|\theta_t - \theta^*\|^2] + \frac{3\gamma_0 K_1^2}{\mu} t^{-2\rho-\beta} + 3K_2\gamma_0^2 t^{-2\beta-r}, \tag{3.29}$$

which holds for $t \geq t_0$.

By unrolling the recursion, we have

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] \leq \quad A_{t,t_0-1}\,\mathbb{E}[\|\theta_{t_0} - \theta^*\|^2] + \frac{3\gamma_0 K_1^2}{\mu}\sum_{i=t_0}^{t} i^{-2\rho-\beta} A_{n,i} + 3K_2\gamma_0^2 \sum_{i=t_0}^{t} i^{-2\beta-r} A_{n,i}, \tag{3.30}$$

where

$$A_{tj} = \begin{cases} \prod_{k=j+1}^{t}\left(1 - \mu\gamma_0 k^{-\beta}\right), & j < t, \\ 1, & j = t. \end{cases}$$

First we discuss the case when $\frac{1}{2} < \beta < 1$. Without loss of generality we assume that t_0 also satisfies $2\rho t_0^{\beta-1} < \frac{\mu\gamma_0}{2}$. Notice that when $\frac{1}{2} < \beta < 1$,

$$\begin{aligned}|A_{tj}| &\le \exp\left(-\mu\gamma_0 \sum_{k=j+1}^{t} k^{-\beta}\right) \\ &\le \exp\left(\mu\gamma_0 j^{-\beta} - \mu\gamma_0 \int_j^t x^{-\beta} dx\right) \\ &= \exp\left(\mu\gamma_0 j^{-\beta} - \frac{\mu\gamma_0\left(t^{1-\beta} - j^{1-\beta}\right)}{1-\beta}\right).\end{aligned}$$

So the first term on the R.H.S. of (3.30) is $O(\exp(-\mu\gamma_0\frac{t^{1-\beta}}{1-\beta}))$. Besides, for the second term on the R.H.S. of (3.30), we have that

$$\sum_{i=t_0}^{t} i^{-\beta-2\rho} A_{ti} = t^{-2\rho}\sum_{i=t_0}^{t} i^{-\beta} A_{ti} + \sum_{i=t_0}^{t-1}\left(\frac{1}{i^{2\rho}} - \frac{1}{(i+1)^{2\rho}}\right)\sum_{j=t_0}^{i} j^{-\beta} A_{tj}. \tag{3.31}$$

For $t_0 \le i \le t$,

$$\sum_{j=t_0}^{i} j^{-\beta} A_{tj} = \frac{1}{\mu\gamma_0}\sum_{j=t_0}^{i}\left(A_{tj} - A_{t,j-1}\right) = \frac{1}{\mu\gamma_0}\left(A_{t,i} - A_{t,t_0-1}\right). \tag{3.32}$$

Notice that

$$\frac{1}{i^{2\rho}} - \frac{1}{(i+1)^{2\rho}} = i^{-2\rho}\left(2\rho i^{-1} + O\left(i^{-2}\right)\right) = 2\rho i^{-2\rho-1} + O\left(i^{-2\rho-2}\right), \tag{3.33}$$

So we have

$$\begin{aligned}\sum_{i=t_0}^{t-1}\left(\frac{1}{i^{2\rho}} - \frac{1}{(i+1)^{2\rho}}\right) A_{ti} &= \sum_{i=t_0}^{t-1}\left(2\rho i^{-2\rho-1} + O\left(i^{-2\rho-2}\right)\right) A_{ti} \\ &\le \frac{\mu\gamma_0}{2}\sum_{i=t_0}^{t} i^{-2\rho-\beta} A_{ti} + O\left(t^{-2\rho-1}\right).\end{aligned} \tag{3.34}$$

The last inequality comes from the choice of t_0 such that $2\rho t_0^{\beta-1} < \frac{\mu\gamma_0}{2}$. Combining (3.32) and (3.34) with (3.31), we have

$$\begin{aligned}\sum_{i=t_0}^{t} i^{-\beta-2\rho} A_{ti} &\le \frac{1}{\mu\gamma_0}\left(A_{t,t} - A_{t,t_0-1}\right) t^{-2\rho} + (\frac{\mu\gamma_0}{2}\sum_{i=t_0}^{t} i^{-2\rho-\beta} A_{ti})(\frac{1}{\mu\gamma_0}(A_{t,t} - A_{t,t_0-1})) \\ &\quad + O\left(t^{-2\rho-1}\right) \\ &\le \frac{1}{\mu\gamma_0} t^{-2\rho} + \frac{1}{2}\sum_{i=t_0}^{t} i^{-2\rho-\beta} A_{ti} + O\left(t^{-2\rho-1}\right)\end{aligned} \tag{3.35}$$

Therefore,

$$\sum_{i=t_0}^{t} i^{-\beta-2\rho} A_{ti} \leq \frac{2}{\mu\gamma_0} t^{-2\rho} + O\left(t^{-2\rho-1}\right). \tag{3.36}$$

Repeating the argument above, we have

$$\begin{aligned}\sum_{i=t_0}^{t} i^{-r-2\beta} A_{ti} \leq& \frac{1}{\mu\gamma_0}\left(A_{t,t} - A_{t,t_0-1}\right) t^{-r-\beta} + \left(\frac{\mu\gamma_0}{2}\sum_{i=t_0}^{t} i^{-r-2\beta} A_{ti}\right)\left(\frac{1}{\mu\gamma_0}(A_{t,t} - A_{t,t_0-1})\right) \\ &+ O\left(t^{-r-\beta-1}\right) \\ \leq& \frac{1}{\mu\gamma_0} t^{-r-\beta} + \frac{1}{2}\sum_{i=t_0}^{t} i^{-2\beta-r} A_{ti} + O\left(t^{-r-\beta-1}\right) \\ \leq& \frac{2}{\mu\gamma_0} t^{-r-\beta} + O\left(t^{-r-\beta-1}\right)\end{aligned} \tag{3.37}$$

Combining (3.36), (3.37) with (3.30), we conclude that $\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] = O(t^{-(2\rho)\wedge(\beta+r)})$ for $t \geq t_0$.

Now we consider the case when $\beta = 1$. For $t_0 - 1 \leq j \leq t$,

$$\begin{aligned} A_{tj} \leq& \exp\left(\mu\gamma_0/j - \mu\gamma_0 \int_j^t x^{-1} dx\right) \\ =& \exp\left(\mu\gamma_0/j - \mu\gamma_0 \ln(t/j)\right) \\ =& \exp\left(\mu\gamma_0/j\right)(j/t)^{\mu\gamma_0} \end{aligned} \tag{3.38}$$

For the second and the third term on the R.H.S. of (3.30), we have

$$\begin{aligned} \sum_{i=t_0}^{t} i^{-2\rho-1} |A_{ti}| \leq& \exp\left(\mu\gamma_0/t_0\right)\left(\sum_{i=t_0}^{t-1} i^{-2\rho-1}(i/t)^{\mu\gamma_0} + t^{-2\rho-1}\right) \\ \leq& \exp\left(\mu\gamma_0/t_0\right) t^{-\mu\gamma_0} \int_{t_0}^{t} x^{\mu\gamma_0-2\rho-1} dx + O(t^{-2\rho-1}) \\ \leq& \exp\left(\mu\gamma_0/t_0\right)\left(\mu\gamma_0 - 2\rho\right)^{-1} t^{-2\rho} + O(t^{-2\rho-1}), \\ \sum_{i=t_0}^{t-1} i^{-2-r} |A_{ti}| \leq& \exp\left(\mu\gamma_0/t_0\right)\sum_{i=t_0}^{t-1} i^{-2-r}(i/t)^{\mu\gamma_0} + O(t^{-2-r}) \\ \leq& \exp\left(\mu\gamma_0/t_0\right)\left(\mu\gamma_0 - r - 1\right)^{-1} t^{-r-1} + O(t^{-2-r}). \end{aligned} \tag{3.39}$$

Therefore, by combining (3.39) with (3.30), we have $\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2] = O(t^{-(2\rho)\wedge(\beta+r)})$ for $t \geq t_0$. (3.27) holds because $g(\cdot)$ is L-smooth. □

If Θ is a closed and convex set of $\mathbb{R}^d$, and the iteration (3.24) changes to $\theta_{t+1} = \mathrm{pr}_\Theta(\theta_t - \gamma_t(\nabla_\theta g(\theta_t) + B_t + V_t))$, the conclusion of Lemma 1 still holds and the discussions are the same as before.

3.B Omitted Proofs of Section 3.3

Proof of Theorem 4

Proof of Theorem 4. Define $B_t = \mathbb{E}[\frac{G_{m_t}(\theta_t+h_tZ,Y_{m_t})-G_{m_t}(\theta_t,Y_{m_t})}{h_t}Z] - \nabla g(\theta_t)$ and

$$V_t = \frac{1}{N_t}\sum_{l=1}^{N_t} \frac{G_{m_t}(\theta_t+h_tZ_{t,l},Y_{m_t,l})-G_{m_t}(\theta_t,Y_{m_t,l})}{h_t}Z_{t,l} - \mathbb{E}[\frac{G_{m_t}(\theta_t+h_tZ,Y_{m_t})-G_{m_t}(\theta_t,Y_{m_t})}{h_t}Z].$$

We want to show that B_t and V_t satisfy conditions of Lemma 1.

Under Assumption 1, 2, and 4, for $k \geq 1$, we have

$$\begin{aligned}
&\| \mathbb{E}[\frac{G_k(\theta+hZ,Y_k)-G_k(\theta,Y_k)}{h}Z] - \nabla g(\theta)\| \\
\leq & \| \mathbb{E}[\frac{G_k(\theta+hZ,Y_k)-G_k(\theta,Y_k)}{h}Z] - \mathbb{E}[\frac{g(\theta+hZ)-g(\theta)}{h}Z]\| \\
&+ \| \mathbb{E}[\frac{g(\theta+hZ)-g(\theta)}{h}Z] - \nabla g(\theta)\|.
\end{aligned}$$

Since

$$\begin{aligned}
&\| \mathbb{E}[\frac{G_k(\theta+hZ,Y_k)-G_k(\theta,Y_k)}{h}Z] - \mathbb{E}[\frac{g(\theta+hZ)-g(\theta)}{h}Z]\| \\
=& \| \mathbb{E}[\mathbb{E}[\frac{G_k(\theta+hZ,Y_k)-G_k(\theta,Y_k)}{h}Z - \frac{g(\theta+hZ)-g(\theta)}{h}Z|Z]]\| \\
=& \| \mathbb{E}[\frac{g_k(\theta+hZ)-g_k(\theta)}{h}Z - \frac{g(\theta+hZ)-g(\theta)}{h}Z]\| \\
\leq & \mathbb{E}[(|\frac{g_k(\theta+hZ)-g(\theta+hZ)}{h}| + |\frac{g_k(\theta)-g(\theta)}{h}|)\|Z\|] \\
\leq & \frac{2}{kh}\mathbb{E}[\|Z\|] \leq \frac{2c_1d^{\frac{1}{2}}}{kh},
\end{aligned}$$

and

$$\begin{aligned}
&\| \mathbb{E}[\frac{g(\theta+hZ)-g(\theta)}{h}Z] - \nabla g(\theta)\| \\
=& \| \mathbb{E}[ZZ^\top \nabla g(\theta+\bar{h}Z) - ZZ^\top \nabla g(\theta)\| \\
\leq & \mathbb{E}[\|ZZ^\top\| \|\nabla g(\theta+\bar{h}Z) - \nabla g(\theta)\|] \\
\leq & L\,\mathbb{E}[\|ZZ^\top\| \|\bar{h}Z\|] \\
\leq & Lh\,\mathbb{E}[\|Z\|^3] \leq c_3Lhd^{\frac{3}{2}},
\end{aligned}$$

where $\bar{h}$ satisfies $0 \leq \bar{h} \leq h$ according to mean value theorem.

Therefore, by setting $m_t = \lceil m_0 d^2\, t^{2\rho} \rceil$ and $h_t = d^{-\frac{3}{2}} t^{-\rho}$, we have

$$\begin{aligned}
&\| \mathbb{E}[\frac{G_{m_t}(\theta_t + h_t Z, Y_{m_t}) - G_{m_t}(\theta_t, Y_{m_t})}{h_t} Z | \theta_t] - \nabla g(\theta_t) \| \\
\leq& \frac{2c_1 d^{\frac{1}{2}}}{m_t h_t} + c_3 L h_t d^{\frac{3}{2}} = O(t^{-\rho}).
\end{aligned}$$

On the other hand, by Assumption 3,

$$\begin{aligned}
\mathbb{E}[\|V_t\|^2 | \theta_t] =& \frac{1}{N_t} \mathbb{E}[\| \frac{G_{m_t}(\theta_t + h_t Z, Y_{m_t}) - G_{m_t}(\theta_t, Y_{m_t})}{h_t} Z \\
&- \mathbb{E}[\frac{G_{m_t}(\theta_t + h_t Z, Y_{m_t}) - G_{m_t}(\theta_t, Y_{m_t})}{h_t} Z] \|^2 | \theta_t] \\
\leq& \frac{1}{N_t} \mathbb{E}[\| \frac{G_{m_t}(\theta_t + h_t Z, Y_{m_t}) - G_{m_t}(\theta_t, Y_{m_t})}{h_t} Z \|^2 | \theta_t] = O(t^{-r}).
\end{aligned}$$

According to Lemma 1, the conclusion of Theorem 4 holds. □

Proof of Theorem 8

Proof of Theorem 8. First, suppose that Assumption 5 (i) holds, so $d^{-2p} k^{-2\rho p}\, \mathbb{E}[C_{m_k}] \to m_0^p \kappa_1$ as $k \to \infty$. The expected cumulative computation cost by the t-th iteration is

$$\begin{aligned}
\mathbb{E}\, T_t &= 2\, \mathbb{E} \sum_{j=1}^{t} N_j C_{m_j} \\
&= 2 \sum_{j=1}^{t} N_j\, \mathbb{E}\, C_{m_j} \\
&= 2N_0 \sum_{j=1}^{t} d^{5-\frac{3}{2}q} j^{r+\rho(2-q)}\, \mathbb{E}\, C_{m_j} \\
&= O(d^{2p+5-\frac{3}{2}q} t^{r+2\rho p+\rho(2-q)+1}).
\end{aligned}$$

According to Theorem 4, $\tau(\epsilon) = O(\epsilon^{-\frac{2}{(2\rho)\wedge(\beta+r)}})$. Therefore,

$$\mathbb{E}\, T_{\tau(\epsilon)} = O(d^{2p+5-\frac{3}{2}q} \epsilon^{-\frac{2(r+2\rho p+\rho(2-q)+1)}{(2\rho)\wedge(\beta+r)}}).$$

Suppose that Assumption 5 (ii) holds. The expected cumulative computation cost by the t-th iteration is

$$\mathbb{E}\, T_t = O(d^{5-\frac{3}{2}q} t^{r+\rho(2-q)+1} \exp((m_0 d^2 + 1) \log(\alpha) t^{2\rho})).$$

According to Theorem 4, $\tau(\epsilon) \leq \kappa^{\frac{1}{(2\rho)\wedge(\beta+r)}} \epsilon^{-\frac{2}{(2\rho)\wedge(\beta+r)}}$. Therefore,

$$\mathbb{E}\, T_{\tau(\epsilon)} = O(d^{5-\frac{3}{2}q} \epsilon^{-\frac{2(r+\rho(2-q)+1)}{(2\rho)\wedge(\beta+r)}} \exp((m_0 d^2 + 1) \kappa^{\frac{2\rho}{(2\rho)\wedge(\beta+r)}} \epsilon^{-\frac{4\rho}{(2\rho)\wedge(\beta+r)}} \log \alpha)).$$

□

Proof of Proposition 9

Proof of Proposition 9. For $\beta \in (\frac{1}{2}, 1]$, we have that

$$\tfrac{r+\rho(2p+2-q)+1}{(2\rho)\wedge(\beta+r)} \geq \tfrac{r+\rho(2p+2-q)+\beta}{(2\rho)\wedge(\beta+r)} = \begin{cases} p+1-\frac{q}{2}+\frac{r+\beta}{2\rho}, & \text{if } 2\rho < \beta + r \\ 1+\frac{\rho(2p+2-q)}{r+\beta}, & \text{if } 2\rho \geq \beta + r \end{cases}$$

Given fixed β, the optimal value of $\frac{r+\rho(2p+2-q)+\beta}{(2\rho)\wedge(\beta+r)}$ is obtained when $2\rho = \beta+r$ and is $p+2-\frac{q}{2}$. Note that the equation $\frac{r+\rho(2p+2-q)+1}{(2\rho)\wedge(\beta+r)} = \frac{r+\rho(2p+2-q)+\beta}{(2\rho)\wedge(\beta+r)}$ holds only if $\beta = 1$. Therefore, $\beta = 1$, $2\rho = r+1$ represents the set of optimal parameters that minimize the computational cost needed for the algorithm to achieve a given precision level.

□

3.C Omitted Proofs of Section 3.4

Proof of Theorem 12

Proof of Theorem 12. According to the proof of Theorem 4,

$$\begin{aligned} &\| \mathbb{E}[\frac{G_{m_t}(\theta_t + h_t Z, Y) - G_{m_t}(\theta_t, Y)}{h_t} Z|\theta_t] - \nabla g(\theta_t)\| \\ \leq &\frac{2c_1 d^{\frac{1}{2}}}{h_t} M^{-m_t\alpha} + c_3 L h_t d^{\frac{3}{2}} = O(t^{-\rho}). \end{aligned}$$

Therefore,

$$\begin{aligned} \|B_t\| &= \| \mathbb{E}[\sum_{k=1}^{m_t} \frac{1}{N_{t,k}} \sum_{\ell=1}^{N_{t,k}} (F_k^{\text{sm}}(\theta_t; h_t, Z_{t,k,l}, Y_{t,k,l}) - F_{k-1}^{\text{sm}}(\theta_t; h_t, Z_{t,k,l}, Y_{t,k,l}))|\theta_t] - \nabla g(\theta_t)\| \\ &= \| \mathbb{E} \frac{G_{m_t}(\theta_t + h_t Z, Y) - G_{m_t}(\theta_t, Y)}{h_t} Z|\theta_t] - \nabla g(\theta_t)\| = O(t^{-\rho}). \end{aligned}$$

On the other hand,

$$
\begin{aligned}
&\mathbb{E}[\|F_k^{\mathrm{sm}}(\theta;h,Z,Y)-F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)-\mathbb{E}\,F_k^{\mathrm{sm}}(\theta;h,Z,Y)+\mathbb{E}\,F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)\|^2]^{1/2}\\
\leq&\,\mathbb{E}[\|F_k^{\mathrm{sm}}(\theta;h,Z,Y)-F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)-\mathbb{E}[F_k^{\mathrm{sm}}(\theta;h,Z,Y)|Z]+\mathbb{E}[F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)|Z]\|^2]^{1/2}\\
&+\mathbb{E}[\|\,\mathbb{E}[F_k^{\mathrm{sm}}(\theta;h,Z,Y)|Z]-\mathbb{E}[F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)|Z]\\
&-\mathbb{E}\,F_k^{\mathrm{sm}}(\theta;h,Z,Y)+\mathbb{E}\,F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)\|^2]^{1/2}\\
=&\,\mathbb{E}[\mathbb{E}[\|F_k^{\mathrm{sm}}(\theta;h,Z,Y)-F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)\\
&-\frac{g_k(\theta+hZ)-g_k(\theta)}{h}Z-\frac{g_{k-1}(\theta+hZ)-g_{k-1}(\theta)}{h}Z\|^2|Z]]^{1/2}\\
&+\mathbb{E}[\|\frac{g_k(\theta+hZ)-g_k(\theta)}{h}Z-\frac{g_{k-1}(\theta+hZ)-g_{k-1}(\theta)}{h}Z\\
&-\mathbb{E}\,F_k^{\mathrm{sm}}(\theta;h,Z,Y)+\mathbb{E}\,F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)\|^2]^{1/2}\\
\leq&\,\mathbb{E}[\|Z\|^2\,\mathbb{E}[\|\frac{G_k(\theta+hZ,Y)-G_{k-1}(\theta+hZ,Y)-g_k(\theta+hZ)+g_{k-1}(\theta+hZ)}{h}\|^2|Z]]^{1/2}\\
&+\mathbb{E}[\|Z\|^2\,\mathbb{E}[\|\frac{G_k(\theta,Y)-G_{k-1}(\theta,Y)-g_k(\theta)+g_{k-1}(\theta)}{h}\|^2|Z]]^{1/2}\\
&+\mathbb{E}[\|\frac{g_k(\theta+hZ)-g_k(\theta)}{h}Z-\frac{g_{k-1}(\theta+hZ)-g_{k-1}(\theta)}{h}Z\\
&-\mathbb{E}\,F_k^{\mathrm{sm}}(\theta;h,Z,Y)+\mathbb{E}\,F_{k-1}^{\mathrm{sm}}(\theta;h,Z,Y)\|^2]^{1/2}\\
\leq&\,\mathbb{E}[\|Z\|^2\,\mathbb{E}[\|\frac{G_k(\theta+hZ,Y)-G_{k-1}(\theta+hZ,Y)-g_k(\theta+hZ)+g_{k-1}(\theta+hZ)}{h}\|^2|Z]]^{1/2}\\
&+\mathbb{E}[\|Z\|^2\,\mathbb{E}[\|\frac{G_k(\theta,Y)-G_{k-1}(\theta,Y)-g_k(\theta)+g_{k-1}(\theta)}{h}\|^2|Z]]^{1/2}\\
&+\mathbb{E}[\|\frac{g_k(\theta+hZ)-g_k(\theta)}{h}Z-\frac{g_{k-1}(\theta+hZ)-g_{k-1}(\theta)}{h}Z\|^2]^{1/2}\\
\leq&\,\mathbb{E}[\frac{\|Z\|^2}{h^2}M^{-2k\eta}]^{1/2}+\mathbb{E}[\frac{4M\|Z\|^2}{h^2}M^{-2k\alpha}]^{1/2}\\
\leq&\,\frac{\tilde{C}'\sqrt{d}M^{-k\eta}}{h},
\end{aligned}
$$

where the last inequality comes from the assumption that $\alpha\geq\eta$.

Under given parameters h_t, $N_{t,k}$ and m_t, we calculate that

$$
\begin{aligned}
\mathbb{E}[\|V_t\|^2|\theta_t]=&\sum_{k=1}^{m_t}\frac{1}{N_{t,k}^2}\sum_{\ell=1}^{N_{t,k}}\mathbb{E}[\|F_k^{\mathrm{sm}}(\theta_t;h_t,Z_{t,k,l},Y_{t,k,l})-F_{k-1}^{\mathrm{sm}}(\theta_t;h_t,Z_{t,k,l},Y_{t,k,l})\\
&-\mathbb{E}[F_k^{\mathrm{sm}}(\theta_t;h_t,Z_{t,k,l},Y_{t,k,l})|\theta_t]+\mathbb{E}[F_{k-1}^{\mathrm{sm}}(\theta_t;h_t,Z_{t,k,l},Y_{t,k,l})|\theta_t]\|^2|\theta_t]\\
\leq&\sum_{k=1}^{m_t}\frac{C'd}{N_0\kappa_t h_t^2}M^{k(1/2-\eta)}\leq\frac{C'}{N_0(M-1)}t^{-r}.
\end{aligned}
$$

According to Lemma 1, the conclusion of Theorem 12 is proved. □

Proof of Theorem 13

Proof of Theorem 13. The expected cumulative computation cost by the t-th iteration is

$$\mathbb{E}\, T_t = 2\, \mathbb{E} \sum_{j=1}^{t} \sum_{k=1}^{m_j} N_{j,k} C_k.$$

The expected cost for the j-th iteration is

$$\begin{aligned} 2\, \mathbb{E} \sum_{k=1}^{m_j} N_{j,k} C_k \leq & 2 \sum_{k=1}^{m_j} \left(1 + N_0 \kappa_j M^{-k(\eta+1/2)}\right) M^k \\ \leq & 2 \sum_{k=1}^{m_j} M^k + 2 \sum_{k=1}^{m_j} N_0 \kappa_j M^{k(1/2-\eta)}. \end{aligned}$$

If $\eta \neq \frac{1}{2}$,

$$\begin{aligned} \mathbb{E} \sum_{k=1}^{m_j} N_{j,k} C_k \leq & \frac{M^{m_j}}{1 - M^{-1}} + N_0 \kappa_j \frac{M^{m_j(1/2-\eta)_+}}{1 - M^{-|1/2-\eta|}} \\ \leq & c(M^{m_j} + d^4 j^{r+2\rho} M^{m_j(1-2\eta)_+}). \end{aligned}$$

According to definition of m_j, $M^{m_j} \leq d^{\frac{2}{\alpha}} j^{\frac{2\rho}{\alpha}}$, so

$$\mathbb{E} \sum_{k=1}^{m_j} N_{j,k} C_k \leq c(d^{\frac{2}{\alpha}} j^{\frac{2\rho}{\alpha}} + d^{4+\frac{2(1-2\eta)_+}{\alpha}} j^{r+2\rho+\frac{2\rho(1-2\eta)_+}{\alpha}}).$$

If $\eta = \frac{1}{2}$,

$$\begin{aligned} \mathbb{E} \sum_{k=1}^{m_j} N_{j,k} C_k \leq & \frac{M^{m_j}}{1 - M^{-1}} + N_0 \kappa_j m_j \\ \leq & c(M^{m_j} + d^4 j^{r_2\rho} m_j^2) \\ \leq & c(d^{\frac{2}{\alpha}} j^{\frac{2\rho}{\alpha}} + d^4 j^{r+2\rho} \log_M(j)^2). \end{aligned}$$

Using the fact that $r = 2\rho - 1$, we have

$$\mathbb{E}\, T_t \leq c_2 \begin{cases} \sum_{j=1}^{t} \left(d^{\frac{2}{\alpha}} j^{\frac{2\rho}{\alpha}} + d^{4+\frac{2(1-2\eta)_+}{\alpha}} j^{4\rho+\frac{2\rho(1-2\eta)_+}{\alpha}-1}\right), & \text{if } \eta \neq 1/2 \\ \sum_{j=1}^{t} \left(d^{\frac{2}{\alpha}} j^{\frac{2\rho}{\alpha}} + d^4 j^{4\rho-1} \log_M(j)^2\right), & \text{if } \eta = 1/2. \end{cases}$$

Hence there exists $c' \in (0, \infty)$ such that

$$\mathbb{E}\, T_t \leq c' \begin{cases} \left(d^{\frac{2}{\alpha}} t^{\frac{2\rho}{\alpha}+1} + d^{4+\frac{2(1-2\eta)_+}{\alpha}} t^{4\rho+\frac{2\rho(1-2\eta)_+}{\alpha}}\right), & \text{if } \eta \neq 1/2 \\ \left(d^{\frac{2}{\alpha}} t^{\frac{2\rho}{\alpha}+1} + d^4 t^{4\rho} \log_M(t)^2\right), & \text{if } \eta = 1/2. \end{cases}$$

According to Theorem 12, $\tau(\epsilon) = O(\epsilon^{-\frac{1}{\rho}})$. Therefore,

$$\mathbb{E}\, T_{\tau(\epsilon)} = \begin{cases} O\left(d^{\frac{2}{\alpha}}\epsilon^{-\frac{2}{\alpha}-\frac{1}{\rho}} + d^{4+\frac{2(1-2\eta)_+}{\alpha}}\epsilon^{-4-\frac{2(1-2\eta)_+}{\alpha}}\right), & \text{if } \eta \neq 1/2 \\ O\left(d^{\frac{2}{\alpha}}\epsilon^{-\frac{2}{\alpha}-\frac{1}{\rho}} + d^{4}\epsilon^{-4}\log_M(\epsilon^{-1})^2\right), & \text{if } \eta = 1/2. \end{cases}$$

If additionally $\alpha > \frac{1}{2}$ and $\rho > \frac{\alpha}{4\alpha+2(1-2\eta)_+-2}$, then $\frac{2}{\alpha} + \frac{1}{\rho} \leq 4 + \frac{2(1-2\eta)_+}{\alpha}$. In this case,

$$\mathbb{E}\, T_{\tau(\epsilon)} = \begin{cases} O(d^4\epsilon^{-4}), & \text{if } \eta > \frac{1}{2}, \\ O(d^4\epsilon^{-4}(\ln(\epsilon^{-1}))^2), & \text{if } \eta = \frac{1}{2}, \\ O(d^{4+\frac{2(1-2\eta)_+}{\alpha}}\epsilon^{-\left(4+\frac{2-4\eta}{\alpha}\right)}), & \text{if } \eta < \frac{1}{2}. \end{cases}$$

□

3.D Additional Results

In this section, we provide some additional results which are not covered in the main text. The additional results are outlined as follows.

In the first subsection, we consider the scenario when the stochastic performance function associated with an approximating system happens to be Lipschitz continuous, and Infinitesimal Perturbation Analysis (IPA)/ Automatic Differentiation (AD) gradient estimators are guaranteed to be unbiased regarding the approximating system (not the original system) and can be constructed at a computation cost that does not increase linearly with the dimension. The associated gradient-based algorithms, based on unbiased IPA estimators, are proved to converge. The algorithms, convergence rates, optimal choices of algorithm parameters and central limit theorems are established.

In the second subsection, we consider the gradient-based algorithm with FD gradient estimator (Algorithm 1). We provide detailed assumptions and proofs of the central limit theorems, given in the main text by Theorem 10 and Theorem 11.

In the third part, we justify that the parameters given in Theorem 13 are indeed the optimal choices to minimize cumulative computational cost.

Simulation Algorithms with IPA/AD/BP Gradient Estimators

In this section, we propose gradient-based simulation-optimization algorithms that take advantage of the infinitesimal perturbation analysis (IPA) gradient estimators, the automatic differentiation (AD) gradient estimators, or the backpropagation (BP) gradient estimators of the approximating systems. All three classes of gradient estimators can enjoy computational efficiency for gradient evaluation at high-dimensional decision variables. For simplicity, we use IPA gradient estimators in this section to represent all three classes. For the approximating system $G_k(\theta, Y_k)$ with index k, the IPA gradient estimator is given by

$$\nabla_\theta G_k(\theta, Y_k). \tag{3.40}$$

The main advantages of IPA gradient estimators are twofold. First, when the system performance function $G_k(\theta, Y_k)$ is differentiable and is Lipschitz continuous in θ, the IPA gradient estimator is unbiased, in the sense that

$$\mathbb{E}[\nabla_\theta G_k(\theta, Y_k)] = \nabla_\theta\, \mathbb{E}[G_k(\theta, Y_k)]. \tag{3.41}$$

Second, when the dominant computation cost is from running the simulation logic (i.e., evaluating $G_k(\theta, Y_k)$ given θ and Y_k), the IPA gradient estimator with respect to a high dimensional variable $\theta \in \mathbb{R}^d$, can be simultaneously obtained from a *single* simulation run of $G_k(\theta, Y_k)$. That is, when the computation cost of a single simulation run of $G_k(\theta, Y_k)$ is C_k, the computation cost of obtaining the gradient vector $\nabla_\theta G_k(\theta, Y_k)$ is given by $R \cdot C_k$, where $R > 1$ is a constant multiplier that does not increase linearly with dimension d; see [43] and [29].

Consider an increasing positive integer sequence $(m_t : t \geq 1)$. At the t-th step, the algorithm updates the θ_{t-1} from the previous step by using a gradient estimator constructed from the m_t-th approximating system. Specifically, the algorithm generates N_t independent copies of the random input Y_{m_t}, noted as $\{Y_{m_t,l}\}_{l=1}^{N_t}$. Correspondingly, the algorithm runs N_t independent simulation copies of the simulation logic of the m_t-th system and obtains N_t copies of the IPA gradient estimator

$$\nabla_\theta G_{m_t}(\theta_{t-1}, Y_{m_t,1}), \nabla_\theta G_{m_t}(\theta_{t-1}, Y_{m_t,2}), \ldots, \nabla_\theta G_{m_t}(\theta_{t-1}, Y_{m_t,N_t}). \tag{3.42}$$

The algorithm then averages the N_t independent IPA gradient estimator as

$$H_t(\theta_{t-1}) := \frac{1}{N_t} \sum_{l=1}^{N_t} \nabla_\theta G_{m_t}\left(\theta_{t-1}, Y_{m_t,l}\right) \tag{3.43}$$

and then updates θ_{t-1} as

$$\theta_t = \mathrm{pr}_\Theta(\theta_{t-1} - \gamma_t H_t(\theta_{t-1})). \tag{3.44}$$

For a set of initialization parameters $\gamma_0, N_0, m_0 > 0$, $\beta, r, \rho \geq 0$, we consider algorithm parameters given by

$$m_t = \lceil m_0\, t^{2\rho} \rceil, \quad N_t = \lceil N_0 t^r \rceil, \quad \gamma_t = \gamma_0 \frac{1}{t^\beta} \tag{3.45}$$

for $t \geq 1$. We impose the following assumption on the sequence of approximating systems.

Assumption 8. *There exists a positive constant M_0 such that for all $k \in \mathbb{N}$ and $\theta \in \mathbb{R}^d$,*

(i) The expected performance g_k is L_k-smooth, and $L^* := \sup_{k \geq 1} L_k < \infty$;

(ii) $G_k(\cdot, Y_k)$ is $\Psi_k(Y_k)$-Lipschitz continuous and $\mathbb{E}[|\Psi_k(Y_k)|^2] < \infty$;

(iii) $\mathbb{E}[\|\nabla_\theta G_k(\theta, Y_k) - \mathbb{E}\, \nabla_\theta G_k(\theta, Y_k)\|^2] \leq M_0$.

Theorem 15. *Suppose that $\beta \in (\frac{1}{2}, 1]$, $r \geq 0$, $\rho, \gamma_0, N_0, m_0 \in (0, \infty)$. If $\beta = 1$, suppose additionally that $\gamma_0 \in (\max\{\frac{2\rho}{\mu}, \frac{1+r}{\mu}\}, \infty)$. Under Assumption 1, 2, and 8, with θ_t defined in the scheme* (3.43) *and* (3.44)*, there exists a $\kappa \in (0, \infty)$ such that for all $t \in \mathbb{N}$,*

$$\mathbb{E}\left[\|\theta_t - \theta^*\|^2\right] \leq \kappa t^{-(2\rho)\wedge(\beta+r)}, \tag{3.46}$$

and

$$\mathbb{E}\left[g(\theta_t) - g(\theta^*)\right] \leq \frac{1}{2} L\kappa\, t^{-(2\rho)\wedge(\beta+r)}, \tag{3.47}$$

where κ only depends on $\beta, r, \rho, \gamma_0, m_0, N_0, L^*, L$, μ and M_0.

Proof of Theorem 15. First, we show that under conditions of Theorem 15,

$$\|\mathbb{E}\left[\nabla_\theta G_{m_t}(\theta, Y_{m_t}) - \nabla_\theta g(\theta)\right]\| \leq (L^* + L + 2) m_t^{-\frac{1}{2}} \leq \frac{L^* + L + 2}{\sqrt{m_0}} t^{-\rho}.$$

Denote $u_n := \frac{1}{\|\nabla_\theta g_n(\theta) - \nabla_\theta g(\theta)\|}(\nabla_\theta g_n(\theta) - \nabla_\theta g(\theta))$. That is, u_n is the unit vector in $\mathbb{R}^d$ that shares the same direction with $\nabla_\theta g_n(\theta) - \nabla_\theta g(\theta)$. For any $h_n > 0$ and $\theta \in \Theta$, we have

$$\begin{aligned} &|\frac{g_n(\theta + h_n u_n) - g_n(\theta)}{h_n} - \nabla_\theta g_n(\theta)^\top u_n| \\ \leq &|(\nabla_\theta g_n(\theta + \xi_n(\theta) h_n u_n) - \nabla_\theta g_n(\theta))^\top u_n| \\ \leq &\|\nabla_\theta g_n(\theta + \xi_n(\theta) h_n u_n) - \nabla_\theta g_n(\theta)\|, \end{aligned}$$

where $\xi_n(\theta) \in (0, 1)$ and its value depends on n and θ. Using the fact that $g_n(\cdot)$ is L_n-smooth, we have

$$|\frac{g_n(\theta + h_n u_n) - g_n(\theta)}{h_n} - \nabla_\theta g_n(\theta)^\top u_n| \leq L_n h_n \leq L^* h_n.$$

So

$$\begin{aligned} &\|\mathbb{E}\left[\nabla_\theta G_n(\theta, Y_n) - \nabla_\theta g(\theta)\right]\| \\ =&|(\nabla_\theta g_n(\theta)) - \nabla_\theta g(\theta))^\top u_n| \\ \leq&|\nabla_\theta g_n(\theta)^\top u_n - \frac{g_n(\theta + h_n u_n) - g_n(\theta)}{h_n}| + |\frac{g_n(\theta + h_n u_n) - g(\theta + h_n u_n)}{h_n}| \\ &+ |\frac{g_n(\theta) - g(\theta)}{h_n}| + |\frac{g(\theta + h_n u_n) - g(\theta)}{h_n} - \nabla_\theta g(\theta)^\top u_n| \\ \leq& L^* h_n + \frac{2}{n h_n} + L h_n. \end{aligned}$$

If we set $h_n = n^{-\frac{1}{2}}$, $\|\mathbb{E}\left[\nabla_\theta G_n(\theta, Y_n) - \nabla_\theta g(\theta)\right]\|$ is bounded by $(L^* + L + 2)n^{-\frac{1}{2}}$. Therefore, we have that $\|\mathbb{E}\left[\nabla_\theta G_{m_t}(\theta, Y_{m_t}) - \nabla_\theta g(\theta)\right]\| \leq (L^* + L + 2) m_t^{-\frac{1}{2}} \leq \frac{L^*+L+2}{\sqrt{m_0}} t^{-\rho}$.

On the other hand, under Assumption 8 (iii),

$$\mathbb{E}[\|\frac{1}{N_t}\sum_{l=1}^{N_t}\nabla_\theta G_{m_t}(\theta_t, Y_{m_t,l}) - \nabla g_{m_t}(\theta_t)\|^2|\theta_t] = \frac{1}{N_t}\mathbb{E}[\|\nabla_\theta G_{m_t}(\theta_t, Y_{m_t}) - \nabla g_{m_t}(\theta_t)\|^2|\theta_t] \leq \frac{M_0}{N_0}t^{-r}.$$

Let $B_t = \mathbb{E}\left[\nabla_\theta G_{m_t}(\theta, Y_{m_t}) - \nabla_\theta g(\theta)\right]$ and $V_t = \frac{1}{N_t}\sum_{l=1}^{N_t}\nabla_\theta G_{m_t}(\theta_t, Y_{m_t,l}) - \nabla g_{m_t}(\theta_t)$. The iteration scheme (3.43) and (3.44) can be written as

$$\theta_{t+1} = \theta_t - \gamma_t(\nabla g(\theta_t) + B_t + V_t),$$

and B_t and V_t satisfy all conditions of Lemma 1, according to the analysis above. Therefore, according to Lemma 1, the conclusion of Theorem 15 holds. □

The cumulative computation cost for the algorithm by the $\tau(\epsilon)$-th iteration, is denoted as

$$T_{\tau(\epsilon)} = R\sum_{j=1}^{\tau(\epsilon)} N_j C_{m_j}. \tag{3.48}$$

Theorem 16. *Under Assumption 5 (i),*

$$\mathbb{E}\,T_{\tau(\epsilon)} = O(\epsilon^{-\frac{2(r+2\rho p+1)}{(2\rho)\wedge(\beta+r)}}).$$

Under Assumption 5 (ii),

$$\mathbb{E}\,T_{\tau(\epsilon)} = O(\epsilon^{-\frac{2(r+1)}{(2\rho)\wedge(\beta+r)}}\exp(\epsilon^{-\frac{4\rho}{(2\rho)\wedge(\beta+r)}}\kappa^{\frac{2\rho}{(2\rho)\wedge(\beta+r)}}(m_0+1)\log\alpha)),$$

in which $O(g(\epsilon))$ denotes a function of ϵ that is bounded by a constant multiplied by $g(\epsilon)$.

Proof of Theorem 16. First, suppose that Assumption 5 (i) holds, so $k^{-2\rho p}\,\mathbb{E}[C_{m_k}] \to m_0^p\kappa_1$ as $k \to \infty$. The expected cumulative computation cost by the t-th iteration is

$$\begin{aligned}\mathbb{E}\,T_t &= \mathbb{E}\,R\sum_{j=1}^{t} N_j C_{m_j}\\ &= R\sum_{j=1}^{t} N_j\,\mathbb{E}\,C_{m_j}\\ &= RN_0\sum_{j=1}^{t} t^r\,\mathbb{E}\,C_{m_j}\\ &= O(t^{r+2\rho p+1}).\end{aligned}$$

According to Theorem 15, $\tau(\epsilon) = O(\epsilon^{-\frac{2}{(2\rho)\wedge(\beta+r)}})$. Therefore,

$$\mathbb{E}\,T_{\tau(\epsilon)} = O(\epsilon^{-\frac{2(r+2\rho p+1)}{(2\rho)\wedge(\beta+r)}}).$$

Suppose that Assumption 5 (i) holds. The expected cumulative computation cost by the t-th iteration is

$$\begin{aligned}\mathbb{E}\, T_t &= R\sum_{j=1}^{t} N_j\, \mathbb{E}\, C_{m_j}\\ &= RN_0\sum_{j=1}^{t} t^r\, \mathbb{E}\, C_{m_j}\\ &= O(t^{r+1}\exp((m_0+1)\log(\alpha)t^{2\rho})).\end{aligned}$$

According to Theorem 15, $\tau(\epsilon) \leq \kappa^{\frac{1}{(2\rho)\wedge(\beta+r)}}\epsilon^{-\frac{2}{(2\rho)\wedge(\beta+r)}}$. Therefore,

$$\mathbb{E}\, T_{\tau(\epsilon)} = O(\epsilon^{-\frac{2(r+1)}{(2\rho)\wedge(\beta+r)}}\exp(\epsilon^{-\frac{4\rho}{(2\rho)\wedge(\beta+r)}}\kappa^{\frac{2\rho}{(2\rho)\wedge(\beta+r)}}(m_0+1)\log\alpha)).$$

□

We next devote our attention to studying the asymptotic distribution of θ_t when t tends to infinity and obtain a central limit theorem (CLT). We then change our lens to the available computation budget C and derive a central limit theorem for the best estimator available with the given budget. Both two results of CLT are under suitable regularity assumptions:

Assumption 9. *$H(\theta) := \nabla^2 g(\theta)$ exists for every $\theta \in \Theta$ and is continuous with respect to θ. Denote $H^* := H(\theta^*)$. All eigenvalues of $H^* - \frac{1+r}{2\gamma_0}I$ have positive real parts.*

Assumption 10. *$\|\theta_n\| < \infty$ a.s. $\forall n$.*

Assumption 11. *θ^* is an asymptotically stable solution of the following ordinary differential equation*

$$\frac{dx(t)}{dt} = -\nabla g(x).$$

Define $D(\theta^*) = \{x_0 : \lim_{t\to\infty} x(t|x_0) = \theta^*\}$, where $x(t|x_0)$ denotes the solution to the ordinary differential equation based on initial condition x_0. There exists a compact $S \subset D(\theta^*)$ such that $\theta_n \in S$ infinitely often for almost all sample points.

Assumption 12. *There exists $\delta > 0$ such that $\sup_{n\in\mathbb{N},\theta\in\Theta}\mathbb{E}[\|\nabla_\theta G_n(\theta, Y_n)\|^{2+\delta}] < \infty$.*

Assumption 13. *There exists a continuous function $\mathcal{E}(\cdot)$ such that $n^{1/2}\left(\nabla_\theta g_n(\theta) - \nabla_\theta g(\theta)\right)$ converges to $\mathcal{E}(\theta)$ uniformly for every $\theta \in \Theta$. Especially, denote $C' := \mathcal{E}(\theta^*)$.*

Assumption 14. *$\mathbb{E}[\sup_{n\in\mathbb{N},\theta\in\Theta}\|\nabla_\theta G_n(\theta, Y_n)\|^2] < \infty$.*

Assumption 15. *There exists a constant $b > 0$ such that*

$$\lim_{n\to\infty}\frac{\mathrm{Var}(C_n)}{n^{p(1-b)}} = 0.$$

Theorem 17. *Denote H^* as the Hessian matrix for $g(\theta)$ at θ^* and $\tilde{H} := H^* - \frac{1+r}{2\gamma_0} I$. Denote $C' := \lim_{n\to\infty} n^{1/2} (\nabla_\theta g_n(\theta^*) - \nabla_\theta g(\theta^*))$. Under suitable regularity assumptions, for the optimal algorithm,*

$$n^{(1+r)/2}(\theta_n - \theta^*) \xrightarrow{d} N(-m_0^{-\frac{1}{2}} \tilde{H}^{-1} C', \Sigma) \quad as \quad n \to \infty$$

where

$$\Sigma = \frac{\gamma_0}{N_0} \int_0^\infty \exp(-\tilde{H}u)\, \mathbb{E}[\nabla_\theta G(\theta^*, Y) \nabla_\theta G(\theta^*, Y)^\top] \exp(-\tilde{H}^\top u) du. \tag{3.49}$$

Theorem 18. *Suppose that all conditions of Theorem 17 are satisfied. Let C be the computation budget and $n(C) := \sup\{n \geq 1 : \sum_{j=1}^n N_j R C_{m_j} \leq C\}$. If $n^{-p}\, \mathbb{E}[C_n] \to \kappa_1$ for some $p > 0$ and $\kappa_1 > 0$, then*

$$(\frac{C}{\kappa_1 N_0 R m_0^p (p+1)})^{1/2(1+p)} (\theta_{n(C)} - \theta^*) \xrightarrow{d} N(-m_0^{-\frac{1}{2}} \tilde{H}^{-1} C', \Sigma) \quad as \quad C \to \infty$$

with Σ defined the same as in Theorem 17.

Proof of Theorem 17. Suppose that Assumption 1, 2, 8, 9, 10, 11, 12, 13 and 14 hold.

Under Assumption 1, 2, 8, 9, 10, and 11, according to [69] Theorem 2.3.1, (see also Theorem 1 of [75] and Proposition 1 of [94]), $\theta_t \to \theta^*$ w.p.1 as $t \to \infty$.

We show that conditions (2.2.1), (2.2.2), and (2.2.3) in [25] hold. First, we observe that

$$\nabla_\theta g(\theta_t) = \nabla_\theta g(\theta^*) + H(\bar{\theta}_t)(\theta_t - \theta^*) = H(\bar{\theta}_t)(\theta_t - \theta^*),$$

where $\bar{\theta}_t$ lies on the line segment between θ_t and θ^*. Then we have

$$\begin{aligned} \theta_{t+1} - \theta^* =& \theta_t - \theta^* - \gamma_t(\nabla_\theta g(\theta_t) + B_t + V_t) \\ =& (I - \gamma_0 t^{-1} H(\bar{\theta}_t))(\theta_t - \theta^*) - \gamma_0 t^{-1} B_t + \gamma_0 t^{-1} \Phi_t V_t \\ =& (I - \gamma_0 t^{-1} H(\bar{\theta}_t))(\theta_t - \theta^*) - \gamma_0 t^{-1-\rho}(t^\rho B_t) + \gamma_0 t^{-1-\rho+\frac{1}{2}}(t^{\rho-\frac{1}{2}} \Phi_t V_t), \end{aligned}$$

where $B_t = \nabla_\theta g_{m_t}(\theta_t) - \nabla_\theta g(\theta_t)$, $\Phi_t = -I$ and $V_t = \frac{1}{N_t} \sum_{l=1}^{N_t} \nabla_\theta G_{m_t}(\theta_t, Y_{m_t,l}) - \nabla_\theta g_{m_t}(\theta_t)$.

Since $\theta_t \to \theta^*$ w.p.1 as $t \to \infty$ and by the continuity of $H(\cdot)$, we have $H(\bar{\theta}_t) \to H(\theta^*)$ w.p.1. According to Assumption 13, $t^\rho B_t = t^\rho(\nabla_\theta g_{\lceil m_0 t^{2\rho} \rceil}(\theta_t) - \nabla_\theta g(\theta_t)) \to m_0^{-\frac{1}{2}} C'$ w.p. 1, so condition (2.2.1) of [25] holds.

Under Assumption 14, by dominated convergence theorem and the fact that $\theta_t \to \theta^*$ w.p.1 as $t \to \infty$,

$$\begin{aligned} t^r\, \mathbb{E}[V_t V_t^\top | \theta_t] =& \frac{t^r}{N_t} \mathbb{E}[(\nabla_\theta G_{m_t}(\theta_t, Y_{m_t}) - \nabla_\theta g_{m_t}(\theta_t))(\nabla_\theta G_{m_t}(\theta_t, Y_{m_t}) - \nabla_\theta g_{m_t}(\theta_t))^\top | \theta_t] \\ \xrightarrow{p}& \frac{1}{N_0} \mathbb{E}[(\nabla_\theta G(\theta^*, Y) - \nabla g(\theta^*))(\nabla_\theta G(\theta^*, Y)^\top - \nabla g(\theta^*))^\top] \\ =& \frac{1}{N_0} \mathbb{E}[\nabla_\theta G(\theta^*, Y) \nabla_\theta G(\theta^*, Y)^\top] \end{aligned}$$

as $t \to \infty$. So condition (2.2.2) of [25] holds.

For $0 < \delta' < \delta/2$, and any $\lambda > 0$, we have

$$\lim_{k\to\infty} \mathbb{E}\left[1_{\left\{ \left\| k^{\frac{r}{2}} V_k \right\|^2 \geq \lambda k \right\}} \left\| k^{\frac{r}{2}} V_k \right\|^2 \right] \quad \leq \lim_{k\to\infty} \sup \left(\frac{\mathbb{E} \left\| k^{\frac{r}{2}} V_k \right\|^2}{\lambda k} \right)^{\delta'/(1+\delta')} \quad \left(\mathbb{E} \left\| k^{\frac{r}{2}} V_k \right\|^{2(1+\delta')} \right)^{1/(1+\delta')}$$

By Burkholder-Davis-Gundy inequality and the triangle inequality on the $L^{\frac{1}{2(1+\delta')}}$ space, there exists a constant $c_{\delta'}$ which only depends on δ', such that

$$\begin{aligned} \left(\mathbb{E} \left\| k^{\frac{r}{2}} V_k \right\|^{2(1+\delta')} \right)^{1/(1+\delta')} &= \left(\mathbb{E} \left\| \frac{1}{N_0 k^{\frac{r}{2}}} \sum_{l=1}^{N_k} (\nabla_\theta G_{m_k}(\theta_k, Y_{m_k,l}) - \nabla_\theta g_{m_k}(\theta_k)) \right\|^{2(1+\delta')} \right)^{1/(1+\delta')} \\ &\leq c_{\delta'} \left(\mathbb{E}[(\frac{1}{N_0^2 k^r} \sum_{l=1}^{N_k} \| \nabla_\theta G_{m_k}(\theta_k, Y_{m_k,l}) - \nabla_\theta g_{m_k}(\theta_k) \|^2)^{(1+\delta')}] \right)^{1/(1+\delta')} \\ &\leq c_{\delta'} \frac{1}{N_0^2 k^r} \sum_{l=1}^{N_k} \left(\mathbb{E}(\| \nabla_\theta G_{m_k}(\theta_k, Y_{m_k,l}) - \nabla_\theta g_{m_k}(\theta_k) \|^{2(1+\delta')} \right)^{1/(1+\delta')} \\ &\leq \frac{c_{\delta'}}{N_0} (\sup_{k\in\mathbb{N}, \theta\in\Theta} \mathbb{E}[\| \nabla_\theta G_k(\theta, Y_k) \|^{2+\delta}])^{1/(1+\delta')} < \infty. \end{aligned}$$

So $\lim_{k\to\infty} \mathbb{E}\left[1_{\left\{ \left\| k^{\frac{r}{2}} V_k \right\|^2 \geq \lambda k \right\}} \left\| k^{\frac{r}{2}} V_k \right\|^2 \right] \to 0$ for every $\lambda > 0$. Therefore, (2.2.3) of [25] holds and CLT is proved. □

Proof of Theorem 18. Suppose that all conditions of Theroem 17 hold and additionally Assumption 15 holds. Since $n^{-p}\,\mathbb{E}\, C_n \to \kappa_1$, we have

$$N_0^{-1} R^{-1} j^{-r} m_j^{-p}\, \mathbb{E}[N_j R C_{m_j}] = N_0^{-1} R^{-1} m_0^{-p} j^{-r-2\rho p}\, \mathbb{E}[N_j R C_{m_j}] \to \kappa_1.$$

For the cumulative computation cost by iteration n, denoted as T_n, we have

$$\frac{\mathbb{E}\, T_n}{\kappa_1 N_0 R m_0^p (p+1) n^{r+1+2\rho p}} \xrightarrow{p} 1 \tag{3.50}$$

as $n \to \infty$. Under Assumption 15, $\frac{\mathrm{Var}(N_n R C_{m_n})}{\mathbb{E}\, T_n} = o(n^{-1-2\rho p b})$, so

$$\sum_{j=1}^{\infty} \frac{\mathrm{Var}(N_j R C_{m_j})}{\mathbb{E}\, T_j} < \infty.$$

According to Kronecker's Law of Large Numbers,

$$\frac{T_n - \mathbb{E}\, T_n}{\mathbb{E}\, T_n} \xrightarrow{p} 0.$$

Therefore,

$$\frac{T_n}{\kappa_1 N_0 R m_0^p (p+1) n^{r+1+2\rho p}} \xrightarrow{p} 1. \tag{3.51}$$

Combining (3.51) with the conclusion of Theorem 17, we have

$$\left(\frac{T_n}{\kappa_1 N_0 R m_0^p (p+1)}\right)^{\frac{r+1}{2(r+1+2\rho p)}} (\theta_n - \theta^*) \xrightarrow{d} N(-\tilde{H}^{-1}C', \Sigma) \quad as \quad n \to \infty$$

Changing T_n to C and n to $n(C)$, and using the fact that $r+1 = 2\rho$, the conclusion in Theorem 18 is derived. □

CLT for Finite Difference Gradient Estimator

Assumption 16. *There exists $\delta > 0$ such that* $\sup_{t\in\mathbb{N},\theta\in\Theta} \mathbb{E}[\|\frac{G_{m_t}(\theta+h_tZ,Y)-G_{m_t}(\theta,Y)}{h_t}Z\|^{2+\delta}] < +\infty$.

Assumption 17. *Denote $H_n(\cdot)$ as the Hessian matrix of $g_n(\cdot)$. There exists a constant $\tilde{L} > 0$ such that $\sup_{n\in\mathbb{N},\theta\in\Theta} \|H_n(\theta)\| < \tilde{L}$.*

Proof of Theorem 10. Suppose that Assumption 1, 2, 9, 10, 11, 12, 13, 14, 16 and 17 hold.

For any $k \in \mathbb{N}$, $G_k(\cdot, Y)$ is Lipschitz continuous. Because of Assumption 12, we set $q = 2$ in the parameter setting.

We show that conditions (2.2.1), (2.2.2), and (2.2.3) in [25] hold. For simplicity, denote $\mathbb{E}_t(\cdot) := \mathbb{E}[\cdot|\theta_t]$, $H_t^{mc}(\theta) = \frac{G_{m_t}(\theta+h_tZ,Y)-G_{m_t}(\theta,Y)}{h_t}Z$, and $F(\theta, Y) = \nabla_\theta G(\theta, Y)$. We want to show that $t^\rho(\mathbb{E}_t[H_t^{mc}(\theta_t)] - \nabla_\theta g(\theta_t)) \xrightarrow{p} \frac{1}{2}\mathbb{E}[ZZ^\top HZ] + m_0^{-\frac{1}{2}}d^{-1}C'$ and $\mathbb{E}_t[(H_t(\theta_t) - \mathbb{E}_t[H_t^{mc}(\theta_t)])(H_t(\theta_t) - \mathbb{E}_t[H_t^{mc}(\theta_t)])^\top] \xrightarrow{p} \Omega$. First, by Taylor expansion we have

$$\begin{aligned}
&\mathbb{E}_t[H_t^{mc}(\theta_t)] - \nabla g(\theta_t) \\
=&\mathbb{E}_t[\frac{g_{m_t}(\theta_t + h_tZ) - g_{m_t}(\theta_t)}{h_t}Z] - \nabla g(\theta_t) \\
=&\mathbb{E}_t[\nabla g_{m_t}(\theta_t)ZZ^\top + \frac{1}{2}h_tZZ^\top H_{m_t}(\theta_t + \xi(m_t, h_t, Z)h_tZ)Z] - \nabla g(\theta_t) \\
=&\nabla g_{m_t}(\theta_t) - \nabla g(\theta_t) + \frac{1}{2}h_t\,\mathbb{E}_t[ZZ^\top H_{m_t}(\theta_t + \xi(m_t, h_t, Z)h_tZ)Z],
\end{aligned}$$

where $\xi(m_t, h_t, Z) \in (0, 1)$ and $H_{m_t}(\cdot)$ is the Hessian matrix of $g_{m_t}(\cdot)$.

Since $\theta_t \to \theta^*$ w.p.1 and $h_t \to 0$ as $t \to \infty$, $H_{m_t}(\theta_t + \xi(m_t, h_t, Z)h_tZ) \to H$ w.p.1 as $t \to \infty$. According to Assumption 17, for any $t \in \mathbb{N}$ and any $\theta \in \mathbb{R}^d$, $\|H_{m_t}(\theta)\| \leq \tilde{L}$. Therefore, $\|ZZ^\top H_{m_t}(\theta_t+\xi(m_t, h_t, Z)h_tZ)Z\| \leq L\|Z\|^3$. By dominated convergence theorem, $\mathbb{E}_t[ZZ^\top H_{m_t}(\theta_t + \xi(m_t, h_t, Z)h_tZ)Z] \xrightarrow{p} \mathbb{E}[ZZ^\top HZ]$ as $t \to \infty$.

On the other hand, $t^\rho(\nabla_\theta g_{\lceil m_0 d^2 t^{2\rho}\rceil}(\theta_t) - \nabla_\theta g(\theta_t)) \to m_0^{-\frac{1}{2}}d^{-1}C'$ w.p.1. So we have

$$t^\rho(\mathbb{E}_t[H_t^{mc}(\theta_t)] - \nabla_\theta g(\theta_t)) \xrightarrow{p} \frac{1}{2}\mathbb{E}[ZZ^\top HZ] + m_0^{-\frac{1}{2}}d^{-1}C'. \tag{3.52}$$

By Taylor expansion $\frac{G_{m_t}(\theta_t+h_tZ,Y)-G_{m_t}(\theta_t,Y)}{h_t}Z = ZZ^\top F_{m_t}(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y)$ with $0 < \lambda(Y,Z,h_t,\theta_t) < 1$. Therefore,

$$\begin{aligned}&\mathbb{E}_t[(H_t^{mc}(\theta_t)-\mathbb{E}_t[H_t^{mc}(\theta_t)])(H_t^{mc}(\theta_t)-\mathbb{E}_t[H_t^{mc}(\theta_t)])^\top]\\ =&\mathbb{E}_t[(Z^\top F_{m_t}(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y))^2ZZ^\top]-\mathbb{E}_t[H_t^{mc}(\theta_t)]\,\mathbb{E}_t[H_t^{mc}(\theta_t)]^\top\\ =&\mathbb{E}_t[(Z^\top F_{m_t}(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y))^2ZZ^\top]-\mathbb{E}_t[H_t^{mc}(\theta_t)]\,\mathbb{E}_t[H_t^{mc}(\theta_t)]^\top.\end{aligned}$$

Since $\mathbb{E}_t[H_t^{mc}(\theta_t)] = \nabla g(\theta_t)+o_p(1)$ and $\nabla g(\theta_t) \xrightarrow{p} \nabla g(\theta^*)=0$, we have

$$\mathbb{E}_t[H_t^{mc}(\theta_t)]\,\mathbb{E}_t[H_t^{mc}(\theta_t)]^\top \xrightarrow{p} 0.$$

By the fact that $\theta_t \to \theta^*$ w.p.1, we have $F_{m_t}(\theta_t+\lambda(Y,Z,h_t.\theta_t)h_tZ,Y) = F(\theta^*,Y)+o_p(1)$. We can bound $\|(Z^\top F_{m_t}(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y))^2ZZ^\top\|$ by

$$(Z^\top F_{m_t}(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y))^2ZZ^\top \le \|Z\|^4 \sup_{n\in\mathbb{N},\theta\in\Theta}\|\nabla_\theta G_n(\theta,Y_n)\|^2.$$

so by Assumption 14 and dominated convergence theorem,

$$\mathbb{E}_t[(Z^\top F(\theta_t+\lambda(Y,Z,h_t,\theta_t)h_tZ,Y))^2ZZ^\top] = \mathbb{E}[(Z^\top F(\theta^*,Y))^2ZZ^\top]+o_p(1).$$

Therefore,

$$\begin{aligned}&\mathbb{E}_t[(H_t^{mc}(\theta_t)-\mathbb{E}_t[H_t^{mc}(\theta_t)])(H_t^{mc}(\theta_t)-\mathbb{E}_t[H_t^{mc}(\theta_t)])^\top]\\ =&\mathbb{E}[(Z^\top F(\theta^*,Y))^2ZZ^\top]+o_p(1).\end{aligned}$$

Under Assumption 16, (2.2.3) of [25] can be checked to stand using the same technique as in the proof of Theorem 17.

Therefore, according to [25], the central limit theorem in Theorem 10 is proved. □

Proof of Theorem 11. Suppose that Assumption 15 holds. Since $n^{-p}\,\mathbb{E}\,C_n \to \kappa_1$, we have $N_0^{-1}R^{-1}d^{-2}j^{-r}m_j^{-p}\,\mathbb{E}[N_jC_{m_j}] = N_0^{-1}d^{-2-2p}m_0^{-p}j^{-r-2\rho p}\,\mathbb{E}[N_jC_{m_j}] \to \kappa_1$. Then, for the cumulative computation cost by iteration n, denoted as T_n, similarly to the proof of Theorem 18, we have

$$\frac{T_n}{2\kappa_1N_0d^{2(p+1)}m_0^p(p+1)n^{r+1+2\rho p}} \xrightarrow{p} 1 \tag{3.53}$$

as $n\to\infty$. Combining (3.53) with the conclusion of Theorem 17, we have

$$(\frac{T_n}{2\kappa_1N_0d^{2(p+1)}m_0^p(p+1)})^{\frac{r+1}{2(r+1+2\rho p)}}(\theta_n-\theta^*) \xrightarrow{d} N(-\tilde{H}^{-1}(\frac{1}{2}\,\mathbb{E}[ZZ^\top H^*Z]+m_0^{-\frac{1}{2}}d^{-1}C'),\Sigma).$$

Changing T_n to C and n to $n(C)$, and using the fact that $r+1=2\rho$, the conclusion in Theorem 11 is derived. □

Optimal Set of Algorithm Parameters for Multilevel FD Estimator

Given parameter r, ρ and β, according to the proof of Theorem 13,

$$\mathbb{E}\, T_t \leq c \begin{cases} (d^{\frac{2}{\alpha}} t^{\frac{2\rho}{\alpha}+1} + d^{4+\frac{2(1-2\eta)_+}{\alpha}} t^{r+1+2\rho+\frac{2\rho(1-2\eta)_+}{\alpha}}), & \text{if } \eta \neq 1/2 \\ (d^{\frac{2}{\alpha}} t^{\frac{2\rho}{\alpha}+1} + d^4 t^{r+1+2\rho} \log_M(t)^2), & \text{if } \eta = 1/2 \end{cases}$$

where c is a constant independent with r, ρ and β.

According to Theorem 12, $\tau(\epsilon) = O(\epsilon^{-\frac{2}{(2\rho)\wedge(\beta+r)}})$. Therefore,

$$\mathbb{E}\, T_{\tau(\epsilon)} = \begin{cases} O\left(d^{\frac{2}{\alpha}} \epsilon^{-(\frac{2\rho}{\alpha}+1)\frac{2}{(2\rho)\wedge(\beta+r)}} + d^{4+\frac{2(1-2\eta)_+}{\alpha}} \epsilon^{-\frac{2(r+1+2\rho+\frac{2\rho(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)}} \right), & \text{if } \eta \neq 1/2 \\ O\left(d^{\frac{2}{\alpha}} \epsilon^{-(\frac{2\rho}{\alpha}+1)\frac{2}{(2\rho)\wedge(\beta+r)}} + d^4 \frac{4}{((2\rho)\wedge(\beta+r))^2} \epsilon^{-\frac{2(r+1+2\rho)}{(2\rho)\wedge(\beta+r)}} \log_M(\epsilon^{-1})^2 \right), & \text{if } \eta = 1/2. \end{cases}$$

Now we suppose that ρ is fixed. When $\eta \neq 1/2$, $(\frac{2\rho}{\alpha}+1)\frac{2}{(2\rho)\wedge(\beta+r)}$ is minimized when $\beta + r \geq 2\rho$. Consider the minimization of $\frac{2(r+1+2\rho+\frac{2\rho(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)}$. For $\beta \in (\frac{1}{2}, 1]$, we have that

$$\frac{r+1+\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)} \geq \frac{r+\beta+\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)} = \begin{cases} 1 + \frac{(1-2\eta)_+}{\alpha} + \frac{r+\beta}{2\rho}, & \text{if } 2\rho < \beta + r \\ 1 + \frac{\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{r+\beta}, & \text{if } 2\rho \geq \beta + r \end{cases}$$

The minimal value of $\frac{r+\beta+\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)}$ is obtained when $2\rho = \beta + r$. Note that the equation $\frac{r+1+\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)} = \frac{r+\beta+\rho(2+\frac{2(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)}$ holds only if $\beta = 1$. Therefore, given fixed ρ and for the case when $\eta \neq 1/2$, $\beta = 1$, $r + 1 = 2\rho$ represents the set of optimal parameters that minimize the computational cost needed for the algorithm to achieve a given precision level.

If $\eta = 1/2$, $\frac{4}{((2\rho)\wedge(\beta+r))^2}$ is minimized when $\beta + r \geq 2\rho$. Given fixed ρ, $\frac{2(r+1+2\rho+\frac{2\rho(1-2\eta)_+}{\alpha})}{(2\rho)\wedge(\beta+r)}$ and $(\frac{2\rho}{\alpha}+1)\frac{2}{(2\rho)\wedge(\beta+r)}$ are minimized when $\beta = 1$ and $r + 1 = 2\rho$. Therefore, when $\eta = 1/2$, the order of expected computation cost is minimized when $r + 1 = 2\rho$ and $\beta = 1$.

Chapter 4

Time-Parallel Simulation

In this chapter, we consider the scenarios when simulation informs real-time decisions that are time-sensitive. For example, in the field of transportation, autonomous vehicles must execute maneuver decisions in real-time based on nearby objects, leaving limited time to conduct a simulation study. Dictated by this specific context, simulation may require completion within a short time slot. However, some simulations are inherently complex and would take too long to run on a single processor or computer within the given time slot. As one feasible solution, parallel simulation breaks down the complex processes or systems into smaller parts that can be processed simultaneously, making it possible to handle them within the time constraints.

In simulation, the time progression is described by the so-called "time advancement function", the treatment of which signifies two classes of simulation methods: event-driven and time-driven. In an event-driven (or discrete-event) simulation, time leaps through distinct points in time, which represents the happening of significant events. See [4, 89, 34]. In a time-driven simulation, time is measured at fixed intervals giving the impression that the system evolves continuously over time, and the system state is updated at equally spaced time-steps. Some examples include [108, 49, 72, 7].

Based on the two classes of simulation methods, there have been developments on two corresponding directions of parallelization techniques. One direction is parallel discrete-event simulation (PDES), which handles the execution of discrete-event simulation programs on parallel processors or threads. See [31, 32, 84, 26, 74] for some of them. Compared with PDES, time-parallel simulation takes a different approach by partitioning the time axis of an intended simulation execution and performing the simulation of the resulting time intervals in parallel. Every logical process manages all state variables, but only for the assigned limited time interval. Afterwards, the results of all intervals are aggregated to create the overall simulation result. See [9, 45, 97, 103]. This has the potential for massive parallelism, as the maximum number of logical processes is determined by the number of possible time intervals, which is only restricted by the granularity of the time representation in the simulation implementation.

In this chapter, we assume that the simulation model itself is time-driven, and focus on algorithm design for time-parallel simulation of stochastic models that can be formulated into a non-homogeneous Markov chain. The basic target we consider is to efficiently estimate the expected performance of this Markov chain over a relative long time period. We presume that the execution time slot for simulation is limited, making it impossible to generate the whole path on a single processor, thus necessitates parallelism. We adopt the framework that the whole time axis of this Markov chain is partitioned into separate time intervals of equal length. Each processor is pre-assigned an initial state and a time interval. During the given execution time slot, it conducts a logical process that simulates the Markov chain on the specific time interval with the assigned initial state. Afterwards, the results of all processors are aggregated to give the estimator for expected performance of the Markov chain. More specifically, to achieve aggregation, we use the simulation results to estimate the transition matrices of the Markov chain at each separate time interval and combine the estimated transition matrices, rather than generate independent paths of entire time period. In our framework of time-parallelism, the efficiency of estimation relies on the assignment policy, which determines the initial state and the time interval for each processor. We provide closed-form optimal processor assignment solutions that minimize the asymptotic mean square error of the estimator, when the resource budget for simulation is large.

In some realistic scenarios, due to the lack of prior knowledge of the Markov chain, the optimal assignment policy is unknown and difficult to pre-determine before we start the simulation. A standard assignment policy, such as assigning equal number of processors for each time interval, can perform poorly when the stochastic system itself is large-scaled. To handle this problem, we develop a two-stage parallel simulation procedure that can efficiently learn and adopt the optimal assignment rule within the execution time slot. The key idea is to divide the time slot into two stages with equally time length: in the first stage, a simple uniform assignment rule is adopted for conducting parallel simulation to gather information of the intrinsic Markov chain; in the second stage, we estimate a near-optimal assignment rule by using the information gathered before, and adopt this improved assignment rule to conduct parallel simulation. We prove that the proposed two-stage procedure can improve the efficiency of the associated parallel simulation algorithm, by reducing the mean square error of the estimator given fixed computational resource.

Further, we develop and analyze a near-optimal assignment policy for time-parallel simulation of a special class of non-homogeneous Markov chains, where the performance metrics depend a lot more on the most recent random variables driving the system than on the initial ones. The motivation is derived from the critical observation that in queueing systems, key performance indicators, such as queue length, are heavily dependent on the last busy cycle [60]. By quantifying this dependency, we further enhance the policy for allocating computational resources in parallel simulations, especially when the simulation model is large scaled.

The rest of this chapter is organized as follows. In Section 4.2, we propose the problem of estimating the performance of a non-homogeneous Markov chain and formulate the basic framework for time-parallel simulation when the execution time slot is limited. In Section 4.3, we prove the influence of assignment policy on both asymptotic and finite-sample performance of our algorithm, and provide closed-form optimal assignment policy. In Section 4.4, we develop a two-stage parallel simulation procedure which learns and applies the optimal assignment policy adaptively. In Section 4.5, we discuss the assignment policy for a special class of Markov chains where the performance metrics are heavily dependent on most recent states. In Section 4.6, we conduct numerical experiments to illustrate theoretical findings.

4.1 Related Work

There have been studies using the time-parallel simulation approach for a variety of applications. For example, it has been used for trace-driven simulations of cache memory systems [45], ATM multiplexers [33], stochastic automata networks [18], air traffic networks [67] and multi-grid partial differential equation (PDE) models [41]. Time-parallel simulation of queues using parallel prefix computation algorithms is described by [42].

In time-parallel simulation, a notable challenge on synchronization is that the final and initial states of adjacent time intervals do not necessarily coincide at the interval boundaries, leading to incorrect state changes. To reduce the cost of achieving state consistency between time intervals, Lin and Lazowska [73] assumes the existence of regeneration points, which are states that keep reoccuring throughout a simulation execution. They designs a parallel simulation algorithm that starts from a regeneration point and continues until the regeneration point is reached again. Afterwards, the traces of the parallel simulations are concatenated to get a correct trace of the simulation over the whole time period. Heidelberger and Stone [45] uses fix-up computations to correct the simulations of those time intervals that are known to have started from incorrect states. Kiesling and Pohl [64] proposes approximate state matching and allows the initial and final states of adjacent time intervals to deviate by an acceptable amount. Nevertheless, the idle time of processors is still unavoidable, which reduces the efficiency of parallel simulation.

In the literature, there has been an increasing attention on analysing and improving parallel simulation algorithms for large-scale stochastic systems. Greenberg, Lubachevsky, and Mitrani [42] converts the problem of simulation of certain kinds of queuing networks to recurrence relations that can be solved parallelly. Thulasidasan and Eidenbenz [95] establishes event-driven queue-based model for traffic network and applies scalable parallelization to simulate the behavior of vehicles. Wang and Hong [99] targets at simulating large-scale inventory systems and proposes a recurrent neural networks (RNN) based approach. Wang, Song, and Hong [100] proposes a fast approximation algorithm for the parallel simulation of large-scale queueing networks. Based on their algorithm, they prove the convergence of the estimator when the queueing network scales up. Hong, Song, and Wang [52] provides a vectorized Euler approximation for simulating queueing networks, and enables simultaneous processing of multiple node tasks.

Parallel computing has greatly extended the reach of ranking & selection (R&S) for simulation optimization; see [26, 79, 76, 81, 82, 51] among others. Luo et al. [76] adapts fully sequential procedures into parallel computing environments. To tackle the synchronization problem, Luo et al. [76] proposes APS procedure that does not require active management of output sequence of simulation results. Pei, Nelson, and Hunter [82] evaluates bisection Parallel Adaptive Survivor Selection (bi-PASS) algorithm for R & S procedure. Zhong et al. [114] proposes modifications to original Paulson's procedure to speed up the selection process in parallel computing environments.

4.2 Problem Formulation

Suppose that the dynamics of a stochastic model are formulated into a possibly non homogeneous discrete-time Markov chain $(X_t : t \geq 0)$. The state space of the Markov chain is finite and denoted as S. S has $|S|$ distinct elements, denoted as $x_1, x_2, \ldots, x_{|S|}$. The transition matrix at time t is P_t. So for arbitrary $x_i, x_j \in S$,

$$P_t(i,j) = P(X_t = x_j | X_{t-1} = x_i).$$

The target is that given initial distribution $\mu_0 \in \mathbb{R}^{|S|}$, we want to estimate $h_T = \mathbb{E}[f(X_T)|X_0 \sim \mu_0]$, where $f : S \to \mathbb{R}$ is an arbitrary function and X_T is the state of the Markov chain at time T.

Suppose that the time for a single machine to simulate the whole path from $t = 0$ to $t = T$ is proportional to the time length T. Consider the scenario that the time window for doing simulation is limited, and only allows us to simulate a path of at most time length τ on a single machine, $\tau < T$. But we are allowed to simultaneously do simulation on N identical machines. We denote $\kappa := \frac{T}{\tau}$, and assume that $N > \kappa|S|$.

We want to design strategy for parallel simulation on N machines. After the end of time slot τ, we collect simulation results of N machines and use simulation results to construct an estimator of h_T, denoted as $\hat{h}_T^{N,\kappa}$. We hope that the estimator $\hat{h}_T^{N,\kappa}$ we construct can minimize the mean square error $\mathbb{E}[\|\hat{h}_T^{N,\kappa} - h_T\|^2]$ as much as possible.

4.3 Baseline Parallel Algorithm and Analysis

In this section, we introduce a baseline algorithm for solving the problem raised in Section 4.2. Denote the τ-step transition matrix as $P_1^{(\tau)}, P_2^{(\tau)}, \ldots, P_\kappa^{(\tau)}$, where

$$P_i^{(\tau)} = \prod_{t=i\tau}^{(i+1)\tau-1} P_t$$

for $i = 0, 1, 2, \ldots, \kappa - 1$. The target of the algorithm is to give independent estimation of each $P_i^{(\tau)}$, by assigning different machines to simulate for different time intervals. We analyse the limiting behavior as well as finit sample behavior of estimator $\hat{h}_T^{N,\kappa}$ proposed by this algorithm.

Algorithm design and central limit theorem

In the baseline parallel simulation algorithm, the N machines are divided into different groups. The groups are denoted as $\mathcal{G}_{i,j},\ i = 0, 1, \ldots, \kappa - 1,\ j = 1, 2, \ldots, |S|$. Therefore there are totally $\kappa|S|$ groups. We assign $N_{i,j} = p_{i,j}N$ machines to group $\mathcal{G}_{i,j}$ ($N_{i,j} \geq 0$), and order them to simulate a certain path with specific starting state. The formal algorithm for parallel simulation is provided in Algorithm 3.

Algorithm 3 Parallel Simulation Algorithm

```
for $i = 0, 1, \ldots, \kappa - 1,\ j = 1, 2, \ldots, |S|$ do
    Assign machines in group $\mathcal{G}_{i,j}$ to simulate path from $t = i\tau$ to $t = (i+1)\tau$ with starting state $x_j$. The results are $\hat{X}_{i+1,j,l}$ for $l = 1, 2, \ldots, N_{i,j}$.
end for
Calculate $\hat{P}_i^{(\tau)}$, for $i = 0, 1, \ldots, \kappa - 1$. The $(j, m)$ entry of $\hat{P}_i^{(\tau)}$ is
    $$\hat{P}_i^{(\tau)}(j, m) = \frac{\sum_{l=1}^{N_{i,j}} \mathbb{1}_{\{\hat{X}_{i+1,j,l}=x_m\}}}{N_{i,j}}.$$
Calculate $\hat{h}_T^{N,\kappa} = \mu_0 \prod_{i=0}^{\kappa-1} \hat{P}_i^{(\tau)} f$.
```

We hope to provide answers to the following questions for the baseline algorithm for parallel simulation:

- What is the limiting behavior and mean square error of the estimator $\hat{h}_T^{N,\kappa}$?
- What are the optimal choices of $p_{i,j}$ for $0 \leq i \leq \kappa - 1$ and $1 \leq j \leq |S|$?
- How does the scale of κ and $|S|$ influence the algorithm performance?

In what follows we present central limit theorem for $\hat{h}_T^{N,\kappa}$, which also helps clarify the optimal choices of $p_{i,j}$:

Theorem 19.

$$\sqrt{N}(\hat{h}_T^{N,\kappa} - \mathbb{E}\, f(X_T)) \xrightarrow{d} \mathcal{N}(0, \sum_{i=0,1,2,\ldots,\kappa-1,j=1,2,\ldots,|S|} \frac{\mu_i^2(x_j)\mathcal{A}_{i,j}}{p_{i,j}})$$

as $N \to \infty$, where

$$\mu_{i\tau}(x_j) = P(X_{i\tau} = x_j | X_0 \sim \mu_0), \tag{4.1}$$

$$\mathcal{A}_{i,j}^{(\tau)} = -(c_{i,j}^{(\tau)})^2 + (u_{i,j}^{(\tau)})^\intercal u_{i,j}^{(\tau)}, \tag{4.2}$$

$$c_{i,j}^{(\tau)} = (P_i^{(\tau)}(j, x_1), P_i^{(\tau)}(j, x_2), \ldots, P_i^{(\tau)}(j, x_{|S|})) \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f,$$

$$u_{i,j}^{(\tau)} = diag\{\sqrt{P_i^{(\tau)}(j, x_1)}, \sqrt{P_i^{(\tau)}(j, x_2)}, \ldots, \sqrt{P_i^{(\tau)}(j, x_{|S|})}\} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f.$$

Since $\hat{P}_i^{(\tau)}$ is unbiased estimator of $P_i^{(\tau)}$ for each i, $\mathbb{E}\, \hat{h}_T^{N,\kappa} = \mathbb{E}\, f(X_T)$. Theorem 19 implies that

$$\lim_{N\to\infty} N\, \mathbb{E}[\|\hat{h}_T^{N,\kappa} - \mathbb{E}\, f(X_T)\|^2] = \lim_{N\to\infty} N \operatorname{Var}(\hat{h}_T^{N,\kappa}) = \sum_{i=0,1,2,\ldots,\kappa-1, j=1,2,\ldots,|S|} \frac{\mu_{i\tau}^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}}.$$

Therefore, the allocation rule will be determined by an optimization problem:

$$\begin{aligned} \min_{p_{i,j}} \quad & \sum_{i=0,1,2,\ldots,\kappa-1, j=1,2,\ldots,|S|} \frac{\mu_{i\tau}^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}} \\ s.t. \quad & \textstyle\sum_{j=1}^{|S|} \sum_{i=0}^{\kappa-1} p_{i,j} = 1, \quad p_{i,j} \geq 0. \end{aligned} \tag{4.3}$$

Since $\hat{P}_i^{(\tau)}$ is unbiased estimator of $P_i^{(\tau)}$ for each i, $\mathbb{E}\, \hat{h}_T^{N,\kappa} = \mathbb{E}\, f(X_T)$. Theorem 19 implies that

$$\lim_{N\to\infty} N\, \mathbb{E}[\|\hat{h}_T^{N,\kappa} - \mathbb{E}\, f(X_T)\|^2] = \lim_{N\to\infty} N \operatorname{Var}(\hat{h}_T^{N,\kappa}) = \sum_{i=0,1,2,\ldots,\kappa-1, j=1,2,\ldots,|S|} \frac{\mu_{i\tau}^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}}.$$

For the asymptotic variance of $\hat{h}_T^{N,\kappa}$ in the baseline algorithm, we investigate its dependence on κ and $|S|$. For simplicity, we assume that the allocation rule is uniform allocation (i.e. $p_{i,j} = \frac{1}{\kappa|S|}$ for every i and j). The dependence upper bound of asymptotic variance on κ and $|S|$ is given by the following proposition.

Proposition 20. *Assume that $f(\cdot)$ is a bounded function, and $p_{i,j} = \frac{1}{\kappa|S|}$ for every i and j. Then as $N \to \infty$,*

$$\operatorname{Var}(\hat{h}_T^{N,\kappa}) = O(\frac{|S|\kappa}{N}).$$

Finite Sample Performance

Suppose that the allocation rule is to assign the machines uniformly. That is, $p_{i,j} = \frac{1}{\kappa|S|}$ for $i = 0, 1, 2, \ldots, \kappa - 1$ and $j = 1, 2, \ldots, |S|$. In this subsection, we present a finite sample concentration result for the estimator $\hat{h}_T^{N,\kappa}$.

Theorem 21. *For all $t > 0$, we have that*

$$P(|\hat{h}_T^{N,\kappa} - \mathbb{E}\, f(X_T)| > t) \leq (|S| \vee e) \exp(-\frac{t^2 N}{2e\|f\|\kappa^2|S|^2}).$$

The proof of Theorem 21 is given in the appendices.

4.4 Two-stage Adaptive Algorithm

In the previous sections, we discuss the scenario when the time window for simulation is limited and only allows us to simulate a path of at most time length τ on a single machine. The baseline method is that we assign each machine to simulate a path of length τ independently, and combine simulation results. Hence how to allocate N machines to simulate with different time periods and different starting states becomes an issue. In the baseline algorithm, the allocation rule is determined before the simulation processes start and cannot be adjusted over time. Given that we do not have prior knowledge with τ-step transition matrices $P_0^{(\tau)}, P_1^{(\tau)}, \ldots, P_{\kappa-1}^{(\tau)}$, one way is to allocate the machines uniformly.

Noticing the limitation of the baseline algorithm, we hope to develop an adaptive algorithm that can adjust the allocation rule dynamically, according to current simulation results. Given the time window τ, we can do parallel simulation in a two-stage setting. In the first stage, we assign each machine to simulate a path of length $\frac{\tau}{2}$. The allocation rule in the first stage is uniform allocation. In the second stage, we assign each machine to simulate a path of length $\frac{\tau}{2}$. The critical insight is that the allocation rule in the second stage is adaptive, and is determined by the gathered simulation results in the first stage. In what follows, we state the two-stage parallel simulation algorithm in more detail. We analyse the limiting behavior of the estimator proposed by the two-stage algorithm, and compare its asymptotic performance with the baseline algorithm.

Formally, we present the two-stage adaptive algorithm in Algorithm 4. The target of the algorithm is to give estimation of each $P_i^{(\frac{\tau}{2})}$ for $0 \leq i \leq 2\kappa - 1$ by using the simulation results collected in two stages. More specifically, the simulation results in the first stage serves as the data source for an estimation of $\mu_{i\frac{\tau}{2}}(x_j)$ and $\mathcal{A}_{i,j}^{(\frac{\tau}{2})}$. Given the preliminary estimation in the first stage, in the second stage we can adopt a better allocation rule for parallel simulation and aggregate all simulation results.

For the two-stage estimator $\hat{h}_T^{\text{Adapt}}$, we present a central limit theorem to analyse the asymptotic performance of the estimator:

Theorem 22.

$$\sqrt{N}(\hat{h}_T^{Adapt} - \mathbb{E}\, f(X_T)) \xrightarrow{d} \mathcal{N}(0, V_{\frac{\tau}{2}}^{op}) \tag{4.5}$$

as $N \to \infty$.

Algorithm 4 Two Stage Adaptive Parallel Simulation Algorithm

```
for $i = 0, 1, \ldots, 2\kappa - 1,\ j = 1, 2, \ldots, |S|$ do
    Assign $\tilde{N}_{1,i,j} = \tilde{q}_{1,i,j} N$ machines to simulate a path from $t = i\tau/2$ to $t = (i+1)\tau/2$ with starting state $x_j$, $i = 0, 1, 2, \ldots, 2\kappa - 1, j = 1, 2, \ldots, |S|$. The simulation results are $\tilde{X}_{i+1,j,l}$ for $l = 1, 2, \ldots, \tilde{N}_{1,i,j}$. Here $\tilde{q}_{1,i,j} = \frac{1}{2\kappa|S|}$ for each $i$ and $j$.
end for
Calculate $\hat{P}_i^{(\frac{\tau}{2})}$, for $i = 0, 1, \ldots, 2\kappa - 1$. The $(j, m)$ entry of $\hat{P}_i^{(\frac{\tau}{2})}$ is
    $$\hat{P}_i^{(\frac{\tau}{2})}(j, m) = \frac{\sum_{l=1}^{\tilde{N}_{1,i,j}} \mathbb{1}_{\{\tilde{X}_{i+1,j,l} = x_m\}}}{\tilde{N}_{1,i,j}}.$$
For each $i = 0, 1, \ldots, 2\kappa - 1$ and $j = 1, 2, \ldots, |S|$, calculate
    $$\hat{\mu}_{i\frac{\tau}{2}} = \mu_0 \prod_{k=0}^{i-1} \hat{P}_k^{(\frac{\tau}{2})}$$
and
    $$\hat{\mathcal{A}}_{i,j}^{(\frac{\tau}{2})} = -(\hat{c}_{i,j}^{(\frac{\tau}{2})})^2 + (\hat{u}_{i,j}^{(\frac{\tau}{2})})^\intercal \hat{u}_{i,j}^{(\frac{\tau}{2})}$$
where
    $$\hat{c}_{i,j}^{(\frac{\tau}{2})} = (\hat{P}_i^{(\frac{\tau}{2})}(j, x_1), \hat{P}_i^{(\frac{\tau}{2})}(j, x_2), \ldots, \hat{P}_i^{(\frac{\tau}{2})}(j, x_{|S|})) \prod_{k=i+1}^{2\kappa-1} \hat{P}_k^{(\frac{\tau}{2})} f$$
and
    $$\hat{u}_{i,j}^{(\frac{\tau}{2})} = \mathrm{diag}\{\sqrt{\hat{P}_i^{(\frac{\tau}{2})}(j, x_1)}, \sqrt{\hat{P}_i^{(\frac{\tau}{2})}(j, x_2)}, \ldots, \sqrt{\hat{P}_i^{(\frac{\tau}{2})}(j, x_{|S|})}\} \prod_{k=i+1}^{2\kappa-1} \hat{P}_k^{(\frac{\tau}{2})} f$$
Solve the optimization problem
    $$\begin{aligned} \min_{q'_{i,j}} \quad & \sum_{i=0,1,2,\ldots,2\kappa-1, j=1,2,\ldots,|S|} \frac{\hat{\mu}^2_{i\frac{\tau}{2}}(x_j) \hat{\mathcal{A}}_{i,j}^{(\frac{\tau}{2})}}{q_{i,j} + q'_{i,j}} \\ s.t. \quad & \textstyle\sum_{j=1}^{|S|} \sum_{i=0}^{2\kappa-1} q'_{i,j} = 1, \quad q'_{i,j} \geq 0. \end{aligned} \qquad (4.4)$$
Calculate $\tilde{N}_{2,i,j} = \tilde{q}'_{i,j} N$ for each $i = 0, 1, \ldots, 2\kappa - 1$ and $j = 1, 2, \ldots, |S|$, where $\tilde{q}'_{i,j}$ is the optimal solution of (4.4).
for $i = 0, 1, \ldots, 2\kappa - 1,\ j = 1, 2, \ldots, |S|$ do
    Assign $\tilde{N}_{2,i,j}$ machines to simulate a path from $t = i\tau/2$ to $t = (i+1)\tau/2$ with starting state $x_j$, $i = 0, 1, 2, \ldots, 2\kappa - 1, j = 1, 2, \ldots, |S|$. The simulation results are $\tilde{X}'_{i+1,j,l}$ for $l = 1, 2, \ldots, \tilde{N}_{2,i,j}$.
end for
Calculate $\hat{Q}_i^{(\frac{\tau}{2})}$, for $i = 0, 1, \ldots, 2\kappa - 1$. The $(j, m)$ entry of $\hat{Q}_i^{(\frac{\tau}{2})}$ is
    $$\hat{Q}_i^{(\frac{\tau}{2})}(j, m) = \frac{\sum_{l=1}^{\tilde{N}_{1,i,j}} \mathbb{1}_{\{\tilde{X}_{i+1,j,l} = x_m\}} + \sum_{l=1}^{\tilde{N}_{2,i,j}} \mathbb{1}_{\{\tilde{X}'_{i+1,j,l} = x_m\}}}{\tilde{N}_{1,i,j} + \tilde{N}_{2,i,j}}.$$
Calculate $\hat{h}_T^{\mathrm{Adapt}} = \mu_0 \prod_{i=0}^{2\kappa-1} \hat{Q}_i^{(\frac{\tau}{2})} f$.
```

where $V^{op}_{\frac{\tau}{2}}$ is the optimal value of the following optimization problem:

$$\begin{aligned} \min_{q'_{i,j}} \quad & \sum_{i=0,1,2,\ldots,2\kappa-1,j=1,2,\ldots,|S|} \frac{\mu^2_{i\frac{\tau}{2}}(x_j)\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}{(2\kappa|S|)^{-1}+q'_{i,j}} \\ s.t. \quad & \textstyle\sum_{j=1}^{|S|}\sum_{i=0}^{2\kappa-1} q'_{i,j}=1, \quad q_{i,j}\geq 0. \end{aligned} \tag{4.6}$$

Here

$$\mathcal{A}^{(\frac{\tau}{2})}_{i,j} = -(c^{(\frac{\tau}{2})}_{i,j})^2 + (u^{(\frac{\tau}{2})}_{i,j})^\intercal u^{(\frac{\tau}{2})}_{i,j},$$

$$c^{(\frac{\tau}{2})}_{i,j} = (P^{(\frac{\tau}{2})}_i(j,x_1), P^{(\frac{\tau}{2})}_i(j,x_2),\ldots,P^{(\frac{\tau}{2})}_i(j,x_{|S|}))\prod_{k=i+1}^{2\kappa-1} P^{(\frac{\tau}{2})}_k f$$

and

$$u^{(\frac{\tau}{2})}_{i,j} = diag\{\sqrt{P^{(\frac{\tau}{2})}_i(j,x_1)},\sqrt{P^{(\frac{\tau}{2})}_i(j,x_2)},\ldots,\sqrt{P^{(\frac{\tau}{2})}_i(j,x_{|S|})}\}\prod_{k=i+1}^{2\kappa-1} P^{(\frac{\tau}{2})}_k f.$$

Furthermore, if the optimal solution of (4.6)*, denoted as $\{q^*_{i,j}\}$, satisfy $q^*_{i,j}>0$ for each $i=0,1,\ldots,2\kappa-1$ and $j=1,2,\ldots,|S|$, then*

$$V^{op}_{\frac{\tau}{2}} = \frac{1}{2}\left(\sum_{i=0,1,2,\ldots,2\kappa-1,j=1,2,\ldots,|S|} \mu_{i\frac{\tau}{2}}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}\right)^2. \tag{4.7}$$

In the two-stage algorithm, we use the simulation results of the first stage to give estimators of $\mu_{i\frac{\tau}{2}}(x_j)$ and $\mathcal{A}^{(\frac{\tau}{2})}_{i,j}$. Then in the second stage, we can adjust the machine allocation rule to lower the empirical variance of $\hat{h}^{N,\kappa}_T$. Compared with one-stage algorithm, two-stage adaptive algorithm optimizes allocation rule. However, the two-stage adaptive algorithm doubles the transition matrices needed to estimate. The reason is that the time window for each stage is now $\frac{\tau}{2}$, so we need to estimate each $\frac{\tau}{2}$-step matrices $P^{(\frac{\tau}{2})}_0, P^{(\frac{\tau}{2})}_1,\ldots,P^{(\frac{\tau}{2})}_{2\kappa-1}$, instead of $P^{(\tau)}_0, P^{(\tau)}_1,\ldots,P^{(\tau)}_{\kappa-1}$. Therefore, whether the two-stage algorithm has better performance or not needs to be investigated. The next theorem illustrates that the two-stage adaptive algorithm can definitely reduce the asymptotic variance of $\hat{h}^{N,\kappa}_T$, compared with one-stage algorithm under uniform allocation rule.

Theorem 23. *Assume that $V^{op}_{\frac{\tau}{2}}$ has the form as in* (4.7)*, and V^{uni}_τ is the asymptotic variance scaled by $\sqrt{N}$ under uniform allocation rule. More specifically,*

$$V^{uni}_\tau = \kappa|S|(\sum_{i=0,1,2,\ldots,\kappa-1,j=1,2,\ldots,|S|} \mu^2_i(x_j)\mathcal{A}_{i,j}).$$

Then

$$V^{op}_{\frac{\tau}{2}} \leq V^{uni}_\tau.$$

The proof of Theorem 23 relies on the following lemma:

Lemma 2. *Assume that $V^{op}_{\frac{\tau}{2}}$ has the form as in (4.7), and assume that $\kappa = 1$, then $V^{op}_{\frac{\tau}{2}} \leq V^{uni}_{\tau}$.*

The proof of Lemma 2 and Theorem 23 are given in the appendices.

4.5 Allocation Rule for A Specific Class of Markov Chains

In this section we focus on the parallel simulation for a special class of non-homogeneous Markov chains. The motivation arises from the domain of queueing systems. In queueing systems, the performance metrics (e.g. queue length) at a specific time instant are heavily dependent on the last busy cycle, i.e., the events that occurred after the queue was empty for the last time. Thus, the performance metrics depend a lot more on the most recent random variables driving the system than on the initial ones. If we can quantify such dependency, it may enable the improvement of the rule of machine allocation in parallel simulation, resulting in a reduction in variance when compared with uniform allocation rules.

For a Markov chain $\{X_t\}$ and time instant T, we define $C(i) = \mathrm{Var}(\mathbb{E}(f(X_T|X_{T-i})))$. $C(i)$ provides one measure of the dependency of X_T on X_{T-i}. One extreme case is that X_T is independent with X_{T-i}, then $C(i) = 0$. Roughly speaking, $C(i)$ is small if X_{T-i} and X_T are "almost" independent. we make the following assumption that characterizes the decay rate of $C(i)$ as i increases. The decay rate can be either polynomial or exponential.

Assumption 18. *Denote $C(i) = \mathrm{Var}(\mathbb{E}(f(X_T)|X_{T-i}))$. There exists constants $c' > 0$ and $\gamma < -1$ independent of T and $|S|$ such that $C(i) \leq c'(i+1)^\gamma$ for $0 \leq i \leq T-1$.*

The analysis in [2] Section IV indicates that $C(i)$ decreases exponentially for a variety of Markov chains, thus satisfying Assumption 18 with arbitrary value of γ. Moreover, we illustrate Assumption 18 by considering a $G_t/D/1/K$ queue. In this queueing system, customers arrive at time step i, $0 \leq i \leq T$. Customers are served by a single server in order of arrival. Service times are all equal to 1. Assume that A_i customers arrive at time step i, where A_i's are independent square-integrable random variables. The capacity of the queue is K. Let X_i be the number of customers waiting in the queue at time-step i for $i = 0, 1, \ldots, T$. Then X_i satisfies the recursive equation

$$X_i = \max(\min(X_{i-1} + A_i - 1, 0), K) \tag{4.8}$$

for $i = 1, 2, \ldots, T$. Furthermore, we assume that the queue is empty at $t = 0$, that is, $X_0 = 0$. We show that in this example, $C(i)$ decreases exponentially with i under certain conditions on the service times.

In more detail, if there exist constant $\alpha > 0$ and $\beta < 1$ independent of T such that

$$\mathbb{E}\, e^{\alpha(A_i - 1)} \leq \beta$$

for $1 \leq i \leq T$, then $C(i) \leq c'\beta^i$ with some constant c'. The reason is that

$$X_T - X_{i,0} \leq \max_{i+1 \leq j \leq T} S_j^+,$$

where $X_{i,0}$ is the number of customers in the queue at time-step T if there are no costumers in the queue at time-step $T - i$, and $S_j^+ = \max(\sum_{k=T-j}^{T-1}(A_i - 1), 0)$ for $1 \leq j \leq T$. Therefore,

$$\begin{aligned} C(i) &\leq \|X_T - X_{i,0}\|^2 \\ &\leq \sum_{j=i+1}^{T} \left\|S_j^+\right\|^2. \end{aligned}$$

On the other hand, since the A_i' 's are independent, $E\left(e^{\gamma S_j}\right) \leq \kappa^j$ for $1 \leq j \leq T$. Furthermore, as $(x^+)^2 \leq 2e^x$ for $x \in \mathbb{R}, \gamma^2 \left(S_j^+\right)^2 \leq 2e^{\gamma S_j}$. Taking expectations implies that $\gamma^2 \left\|S_j^+\right\|^2 \leq 2\kappa^j$, and so $C(i) \leq \gamma'\kappa^i$, where $\gamma' = 2\gamma^{-2}/(1-\kappa)$.

Under Assumption 18, we present a parallel simulation algorithm as follows, which is a modification of baseline parallel algorithm (Algorithm 3).

- Assign $N_{i,j} = p_{i,j}N$ machines to simulate path from $t = i\tau$ to $t = (i+1)\tau$ with starting state x_j, where

  $$p_{i,j} = \frac{(\kappa - i)^{\frac{\gamma-1}{2}}}{|S| \sum_{k=1}^{\kappa} k^{\frac{\gamma-1}{2}}}.$$

  The results are $\hat{X}_{i+1,j,l}$ for $l = 1, 2, \ldots, N_{i,j}$.

- Calculate $\hat{P}_i^{(\tau)}$, for $i = 0, 1, \ldots, \kappa - 1$. The (j, m) entry of $\hat{P}_i^{(\tau)}$ is

  $$\hat{P}_i^{(\tau)}(j, m) = \frac{\sum_{l=1}^{N_{i,j}} \mathbb{1}_{\{\hat{X}_{i+1,j,l} = x_m\}}}{N_{i,j}}.$$

- Calculate $\hat{h}_T^{N,\kappa} = \mu_0 \prod_{i=0}^{\kappa-1} \hat{P}_i^{(\tau)} f$.

In this algorithm, we adjust the allocation rule so that the number of machines targeting to simulate path from $i\tau$ to $(i+1)\tau$ is proportional to $(\kappa - i)^{\frac{\gamma-1}{2}}$. Therefore, more machines are allocated to simulate paths closer to T. Proposition 24 below shows that this algorithm eliminates the dependence on κ for asymptotic variance of $\hat{h}_T^{N,\kappa}$.

Proposition 24. *Under Assumption 18, as $N \to \infty$,*

$$\mathrm{Var}(\hat{h}_T^{N,\kappa}) = O(\frac{|S|}{N}).$$

The proof of Proposition 24 is provided in the appendices. Recall that in Proposition 20, the asymptotic variance has the order $O(\frac{\kappa|S|}{N})$. Therefore, the modification on allocation rule does reduce the order of asymptotic variance when the time horizon T is long and thus improves estimation efficiency.

4.6 Numerical Experiments

In this section, we implement the two algorithms: Algorithm 3 (with a uniform allocation rule) and Algorithm 4 (with two-stage adaptive allocation rules) on numerical experiments. We find that the experiment results support our theoretical findings.

We consider simulating a $G_t/D/1/K$ queue. Specifically, in the queueing system, customers arrive at discrete time step i, $0 \le i \le T$. Customers are served by a single server in order of arrival. Service times are all equal to 1. A_i customers arrive at time step i, where A_i's are independent random variables and follow Poisson distribution with parameter $\lambda_i = 1.5 + 0.5\sin(i\pi/T)$. The capacity of the queue is $K = 20$. Let X_i be the number of customers waiting in the queue at time-step i for $i = 0, 1, \ldots, T$. Then X_i satisfies the recursive equation

$$X_i = \max(\min(X_{i-1} + A_i - 1, 0), K) \tag{4.9}$$

for $i = 1, 2, \ldots, T$. Furthermore, we assume that the queue is empty at $t = 0$, and simply set $f(x) = x$. Figure 4.1 shows the mean square error of the two simulation algorithms given different number of machines. It also shows that the two-stage adaptive algorithm reduces the mean square error of the estimator, thus having better finite-sample performance compared with using a single uniform allocation rule.

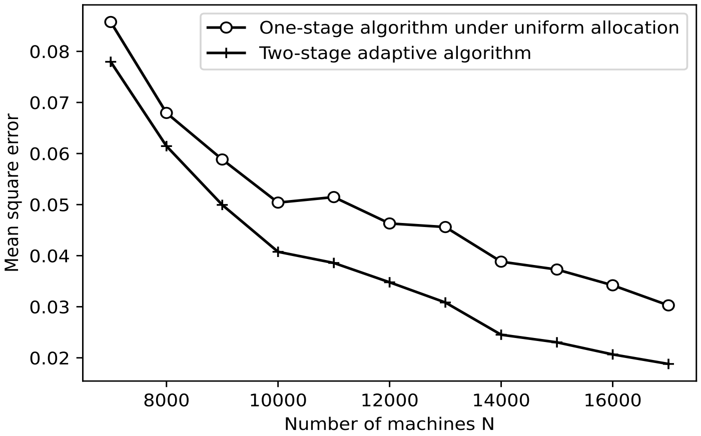


Figure 4.1: Mean square error of $\hat{h}_T^{N,\tau}$ and $\hat{h}_T^{\text{Adapt}}$ for $G_t/D/1/K$ queueing model with $T = 20$ and $K = 20$

Appendices

4.A Omitted Proofs of Section 4.3

Proof of Theorem 19

Proof of Theorem 19. We write $\hat{P}_i^{(\tau)}$ as

$$\hat{P}_i^{(\tau)} = P_i^{(\tau)} + \mathcal{E}_i.$$

$\mathcal{E}_i$ satisfies $\mathbb{E}\,\mathcal{E}_i = 0$. Furthermore, $\mathcal{E}_{i_1}$ and $\mathcal{E}_{i_2}$ are independent for any $i_1 \neq i_2$.

The estimator of $\mathbb{E}\, f(X_T)$ is

$$\hat{h}_T^{N,\kappa} = \mu_0 \prod_{i=0}^{\kappa-1} \hat{P}_i^{(\tau)} f = \mu_0 \prod_{i=0}^{\kappa-1} (P_i^{(\tau)} + \mathcal{E}_i) f,$$

For each i, we have

$$\begin{aligned}\mathbb{E}\,\|\mathcal{E}_i\|_2 &\leq \mathbb{E}\,\|\mathcal{E}_i\|_F \\ &= \mathbb{E}(\sum_{j=1}^{|S|}\sum_{m=1}^{|S|}(\hat{P}_i^{(\tau)}(j,m) - P_i^{(\tau)}(j,m))^2)^{\frac{1}{2}} \\ &\leq (\frac{|S|}{N_{i,j}})^{\frac{1}{2}} = (\frac{|S|}{Np_{i,j}})^{\frac{1}{2}}.\end{aligned} \tag{4.10}$$

Therefore, as $N \to \infty$,

$$\hat{h}_T^{N,\kappa} = \mu_0 \prod_{i=0}^{\kappa-1} P_i^{(\tau)} f + \sum_{i=0}^{\kappa-1} \mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_i \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f + O_p(\frac{1}{N}) \tag{4.11}$$

Since for $\mathcal{E}_i$, we can calculate that

$$\begin{aligned}&\mathbb{E}\,\mathcal{E}_i(j_1,m_1)\mathcal{E}_i(j_2,m_2) = 0 \quad \text{for} \quad j_1 \neq j_2, \\ &\mathbb{E}\,\mathcal{E}_i(j,m_1)\mathcal{E}_i(j,m_2) = -\frac{P_i^{(\tau)}(j,m_1)P_i^{(\tau)}(j,m_2)}{N_{i,j}} \quad \text{for} \quad m_1 \neq m_2, \\ &\mathbb{E}\,\mathcal{E}_i(j,m_1)\mathcal{E}_i(j,m_2) = \frac{P_i^{(\tau)}(j,m)(1-P_i^{(\tau)}(j,m))}{N_{i,j}} \quad \text{for} \quad m_1 = m_2 = m,\end{aligned}$$

then

$$\mathbb{E}(\mathcal{E}_i^\intercal \mu_i^\intercal \mu_i \mathcal{E}_i) = \sum_{j=1}^{|S|} \mu_i^2(x_j)(\text{diag}\{P_i^\tau(j,x_1), P_i^\tau(j,x_2), \ldots, P_i^\tau(j,x_{|S|})\} - v_{ij}v_{ij}^\intercal).$$

Here $v_{ij} = (P_i^\tau(j,x_1), P_i^\tau(j,x_2), \ldots, P_i^\tau(j,x_{|S|}))^\intercal$. By central limit theorem,

$$\sqrt{N}\mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_i \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f = \mathcal{N}(0, \sum_{j=1,2,\ldots,|S|} \frac{\mu_i^2(x_j)\mathcal{A}_{i,j}}{p_{i,j}}) + o_p(1).$$

Because $\mathcal{E}_{i_1}$ and $\mathcal{E}_{i_2}$ are independent for any $i_1 \neq i_2$, Theorem 19 is proved. □

Proof of Proposition 20

Proof of Proposition 20. To analyse the asymptotic variance's dependence on κ and $|S|$, we introduce a different parallel simulation algorithm as below. This algorithm is similar to sample avearge approximation (SAA) method, and aims to generate independent paths. We compare the asymptotic variance of this algorithm with the baseline algorithm above, and show they are the same order of κ and $|S|$.

Algorithm (Independent paths generation):

- Divide the N machines into $\frac{N}{\kappa|S|}$ groups, denoted as $\mathcal{L}_k$ for $k = 1, 2, \ldots, \frac{N}{\kappa|S|}$. Therefore each group has $\kappa|S|$ machines.

- For each group $\mathcal{L}_k$, the machines are marked as $\mathcal{M}_{i,j,k}$, $i = 0, 1, \ldots, \kappa$, $j = 1, 2, \ldots, |S|$. The machine $\mathcal{M}_{i,j,k}$ simulates a path from $t = i\tau$ to $t = (i+1)\tau$, with starting date x_j. The simulation result of $\mathcal{M}_{i,j,k}$ is denoted as $\hat{X}_{i+1,j,k}$. Denote a map $\mathcal{M}_{i,k} : S \to S$ such that $\mathcal{M}_{i,k}(x_j) = \hat{X}_{i+1,j,k}$ for each j.

- Generate $\frac{N}{\kappa|S|}$ independent samples from the distribution μ_0, denoted as $\{x_{0,k}\}_{k=1}^{\frac{N}{\kappa|S|}}$.

- For each $k = 1, 2, \ldots, \frac{N}{\kappa|S|}$ calculate

$$\hat{X}_{T,k} = \mathcal{M}_{\kappa-1,k} \circ \ldots \circ \mathcal{M}_{2,k} \circ \mathcal{M}_{1,k} \circ \mathcal{M}_{0,k}(x_{0,k})$$

The estimator for h_T is given by a sample average: $\hat{h}_{N,\kappa}^{\text{IND}} = \frac{\kappa|S|}{N} \sum_{k=1}^{\frac{N}{\kappa|S|}} f(\hat{X}_{T,k})$.

For this algorithm, it generates $\frac{N}{\kappa|S|}$ independent and unbiased realizations of x_T, thus the variance of $\hat{h}_{N,\kappa}^{\text{IND}}$ is $V^{\text{IND}} = \frac{\kappa|S|\,\text{Var}(f(X_T))}{N}$. Now we compare V^{IND} with the asymptotic variance of the baseline algorithm. We hope to prove that the asymptotic variance of $\hat{h}_{N,\kappa}^{\text{IND}}$ is always larger than the asymptotic variance of $\hat{h}_T^{N,\kappa}$, given uniform allocation rule. More specifically, we want to prove

$$\kappa|S|\operatorname{Var}(f(X_T)) \geq \kappa|S|(\sum_{i=0,1,2,\ldots,\kappa-1,j=1,2,\ldots,|S|} \mu_i^2(x_j)\mathcal{A}_{i,j}), \tag{4.12}$$

since the right side of (4.12) is the asymptotic variance scaled by $\sqrt{N}$ given by uniform allocation rule. To prove this, we use mathematical induction. When $\kappa = 1$, we have

$$\operatorname{Var}(f(X_T)) = \sum_{j=1,2,\ldots,|S|} (f(x_j) - \mu_0 P_1^\tau f)^2 P(X_T = x_j|X_0 \sim \mu_0) \geq \sum_{j=1,2,\ldots,|S|} \mu_0^2(x_j)\mathcal{A}_{0,j}.$$

Suppose that (4.12) holds when $\kappa = K$. When $\kappa = K+1$, we have

$$\operatorname{Var}(f(X_T)) = \mathbb{E}[\operatorname{Var}(f(X_T|X_{T-\tau}))] + \operatorname{Var}(\mathbb{E}[f(X_T|X_{T-\tau})]).$$

By replacing f by $P_K^{(\tau)} f$, we have

$$\operatorname{Var}(\mathbb{E}[f(X_T|X_{T-\tau})]) \geq \sum_{i=0,1,2,\ldots,K-1,j=1,2,\ldots,|S|} \mu_i^2(x_j)\mathcal{A}_{i,j}$$

Some computation yields that

$$\mathbb{E}[\operatorname{Var}(f(X_T|X_{T-\tau}))] \geq \sum_{j=1,2,\ldots,|S|} \mu_K^2(x_j)\mathcal{A}_{K,j}.$$

Therefore, (4.12) holds when $\kappa = K$. By induction (4.12) holds for every $\kappa > 0$.

Because $V^{\mathrm{IND}} = \frac{\kappa|S|\operatorname{Var}(f(X_T))}{N}$, and $\operatorname{Var}(f(X_T)) \leq \|f\|_\infty^2 < \infty$, we conclude that $\operatorname{Var}(\hat{h}_T^{N,\kappa}) = O(\frac{|S|\kappa}{N})$. □

Proof of Theorem 21

Proof of Theorem 21. For a $m \times n$ matrix A, its Schatten $p-$norm is defined as

$$\|A\|_p = (\sum_{i=1}^{\min\{m,n\}} \sigma_i^p(A))^{\frac{1}{p}},$$

where $\{\sigma_i(A)\}_{i=1}^{\min\{m,n\}}$ are singular values of A. When $p = 2$, the Schatten 2-norm of matrix yields the Frobenius norm. For a random matrix X and parameters $p, q \geq 1$, its $L_q(S_p)$ norm is defined as

$$|||X|||_{p,q} := \|X\|_{L_q(S_p)} := \left(\mathbb{E}\|X\|_p^q\right)^{1/q}.$$

By the monotonicity of Schatten norm, for any $p \geq 2$, we have $|||X|||_{p,2} \leq |||X|||_{2,2} = (\mathbb{E}\,\|X\|_F^2)^{\frac{1}{2}}$. To make use of the conclusion in [55] Remark 5.7, we only need to bound $\mathbb{E}\,\|\hat{P}_i^{(\tau)} - P_i^{(\tau)}\|_F^2$ for $i = 0, 1, \ldots, \kappa-1$.

We have

$$\begin{aligned}\mathbb{E}\,\|\hat{P}_i^{(\tau)} - P_i^{(\tau)}\|_F^2 &= \mathbb{E}(\sum_{j=1}^{|S|}\sum_{m=1}^{|S|}(\hat{P}_i^{(\tau)}(j,m) - P_i^{(\tau)}(j,m))^2)\\ &= \sum_{j=1}^{|S|}\sum_{m=1}^{|S|}\mathbb{E}(\hat{P}_i^{(\tau)}(j,m) - P_i^{(\tau)}(j,m))^2\\ &= \sum_{j=1}^{|S|}\sum_{m=1}^{|S|}\frac{P_i^{(\tau)}(j,m)(1-P_i^{(\tau)}(j,m))}{Np_{i,j}}\\ &\le \frac{|S|^2\kappa}{N}.\end{aligned} \tag{4.13}$$

We define $v := \sum_{i=0}^{\kappa-1}\frac{|S|^2\kappa}{N} = \frac{|S|^2\kappa^2}{N}$. Also note that $\hat{P}_0^{(\tau)}, \hat{P}_1^{(\tau)}, \ldots, \hat{P}_{\kappa-1}^{(\tau)}$ are an independent sequence and $\|\hat{P}_i^{(\tau)}\| \le 1$ almost surely for each i. We can use [55] Remark 5.7 to conclude that

$$P(\|\prod_{i=0}^{\kappa-1}\hat{P}_i^{(\tau)} - \prod_{i=0}^{\kappa-1}P_i^{(\tau)}\| \ge t) \le (|S| \vee e)\exp(-\frac{t^2}{2ev})$$

for all $t \ge 0$. Therefore,

$$\begin{aligned}P(|\hat{h}_T^{N,\kappa} - \mathbb{E}\,f(X_T)| \ge t) &= P(|\mu_0\prod_{i=0}^{\kappa-1}\hat{P}_i^{(\tau)}f - \mu_0\prod_{i=0}^{\kappa-1}P_i^{(\tau)}f| \ge t)\\ &\le P(\|\mu_0\|\|\prod_{i=0}^{\kappa-1}\hat{P}_i^{(\tau)} - \prod_{i=0}^{\kappa-1}P_i^{(\tau)}\|\|f\| \ge t)\\ &= P(\|\prod_{i=0}^{\kappa-1}\hat{P}_i^{(\tau)} - \prod_{i=0}^{\kappa-1}P_i^{(\tau)}\| \ge \frac{t}{\|f\|})\\ &\le (|S| \vee e)\exp(-\frac{t^2N}{2e\|f\|\kappa^2|S|^2}).\end{aligned} \tag{4.14}$$

□

4.B Omitted Proofs of Section 4.4

Proof of Theorem 22

Proof of Theorem 22. First we prove that

$$\tilde{q}'_{i,j} = q^*_{i,j} + o_p(1).$$

Then we write each $\hat{Q}_i^{(\frac{\tau}{2})}$ as

$$\hat{Q}_i^{(\frac{\tau}{2})} = P_i^{(\frac{\tau}{2})} + \mathcal{E}_{i,1} + \mathcal{E}_{i,2},$$

where

$$\begin{aligned}\mathcal{E}_{i,1}(j,m) &= \frac{\sum_{l=1}^{\tilde{N}_{1,i,j}}(\mathbb{1}_{\{\tilde{X}_{i+1,j,l}=x_m\}} - P_i^{\frac{\tau}{2}}(j,m))}{\tilde{N}_{1,i,j}+\tilde{N}_{2,i,j}} \\ &= \frac{\sum_{l=1}^{\tilde{N}_{1,i,j}}(\mathbb{1}_{\{\tilde{X}_{i+1,j,l}=x_m\}} - P_i^{\frac{\tau}{2}}(j,m))}{\tilde{N}_{1,i,j}} \frac{(2\kappa|S|)^{-1}}{(2\kappa|S|)^{-1}+q^*_{i,j}}(1+\frac{q^*_{i,j}-\tilde{q}'_{i,j}}{(2\kappa|S|)^{-1}+\tilde{q}'_{i,j}})\end{aligned}$$

and

$$\begin{aligned}\mathcal{E}_{i,2}(j,m) &= \frac{\sum_{l=1}^{\tilde{N}_{2,i,j}}(\mathbb{1}_{\{\tilde{X}'_{i+1,j,l}=x_m\}} - P_i^{\frac{\tau}{2}}(j,m))}{\tilde{N}_{1,i,j}+\tilde{N}_{2,i,j}} \\ &= \frac{\sum_{l=1}^{\tilde{N}_{2,i,j}}(\mathbb{1}_{\{\tilde{X}'_{i+1,j,l}=x_m\}} - P_i^{\frac{\tau}{2}}(j,m))}{\tilde{N}_{2,i,j}} \frac{\tilde{q}'_{i,j}}{(2\kappa|S|)^{-1}+\tilde{q}'_{i,j}}.\end{aligned}$$

We prove that $\|\mathcal{E}_{i,1}\|_2 = O(\frac{1}{\sqrt{N}})$ and $\|\mathcal{E}_{i,2}\|_2 = O(\frac{1}{\sqrt{N}})$ for each i, thus we have

$$\hat{h}_T^{N,\kappa} = \mu_0 \prod_{i=0}^{\kappa-1} P_i^{(\tau)} f + \sum_{i=0}^{\kappa-1} \mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_{i,1} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f + \sum_{i=0}^{\kappa-1} \mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_{i,2} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f + o_p(\frac{1}{\sqrt{N}}) \tag{4.15}$$

as $N \to \infty$. Conditioned on $\{\{\tilde{X}_{i+1,j,l}\}_{l=1,2,\ldots,\tilde{N}_{1,i,j}}\}_{i=0,1,\ldots,\kappa-1,j=1,2,\ldots,|S|}$, each $\mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_{i,2} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f$ is independent and

$$\sqrt{N}\mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_{i,2} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f \xrightarrow{d} \mathcal{N}(0, \sum_{j=1,2,\ldots,|S|} \frac{q^*_{i,j}\mu^2_{i\frac{\tau}{2}}(x_j)\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}{((2\kappa|S|)^{-1}+q^*_{i,j})^2})$$

as $N \to \infty$. On the other hand,

$$\sqrt{N}\mu_0 \prod_{k=0}^{i-1} P_k^{(\tau)} \mathcal{E}_{i,1} \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f \xrightarrow{d} \mathcal{N}(0, \sum_{j=1,2,\ldots,|S|} \frac{(2\kappa|S|)^{-1}\mu^2_{i\frac{\tau}{2}}(x_j)\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}{((2\kappa|S|)^{-1}+q^*_{i,j})^2}).$$

Therefore, (4.5) can be proved by a use of dominated convergence theorem.

If $q^*_{i,j} > 0$ for each $i = 0, 1, \ldots, 2\kappa - 1$ and $j = 1, 2, \ldots, |S|$, then we can write the explicit form for each $q^*_{i,j}$:

$$q^*_{i,j} = \frac{2\sqrt{\mu^2_{i\frac{\tau}{2}}(x_j)\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}}{\sum_{i=0,1,\ldots,2\kappa-1,j=1,2\ldots,|S|}\sqrt{\mu^2_{i\frac{\tau}{2}}(x_j)\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}} - (2\kappa|S|)^{-1}.$$

Then the optimal value of minimization problem (4.6) is (4.7). □

Proof of Lemma 2

Proof of Lemma 2. Under the case $\kappa = 1$, we simplify some notations. We use P_0 to denote $P_0^{(\frac{\tau}{2})}$ and P_1 to denote $P_1^{(\frac{\tau}{2})}$. Therefore, the τ-step transition matrix $P_0^{(\tau)}$ satisfies $P_0^{(\tau)} = P_0 P_1$. For the purpose of simplification we use R to denote $P_0^{(\tau)}$.

If $\kappa = 1$, V_τ^{uni} has the form

$$\begin{aligned}
V_\tau^{\text{uni}} =& |S| \sum_{j=1,2,\dots,|S|} \mu_0^2(x_j) \left(-(\sum_{m=1}^{|S|} R(j,m) f(x_m))^2 + \sum_{m=1}^{|S|} R(j,m) f^2(x_m) \right) \\
=& |S| \sum_{j=1,2,\dots,|S|} \mu_0^2(x_j) (\sum_{m=1}^{|S|} R(j,m)(1 - R(j,m)) f^2(x_m) \\
& - 2 \sum_{m=1}^{|S|} \sum_{l=m+1}^{|S|} R(j,m) R(j,l) f(x_m) f(x_l)) \\
=& |S| \sum_{j=1,2,\dots,|S|} \mu_0^2(x_j) (\sum_{m=1}^{|S|} R(j,m) (\sum_{l \in S, l \neq m} R(j,l)) f^2(x_m) \\
& - 2 \sum_{m=1}^{|S|} \sum_{l=m+1}^{|S|} R(j,m) R(j,l) f(x_m) f(x_l)) \\
=& |S| \sum_{j=1,2,\dots,|S|} \mu_0^2(x_j) \left(\sum_{m \in S, l \in S, m < l} R(j,m) R(j,l) (f(x_m) - f(x_l))^2 \right).
\end{aligned} \tag{4.16}$$

For each $\mathcal{A}_{0,j}^{(\frac{\tau}{2})}$, we can simplify it to be

$$\begin{aligned}
\mathcal{A}_{0,j}^{(\frac{\tau}{2})} =& -(\sum_{k=1}^{|S|} P_1(j,k)(\sum_{m=1}^{|S|} P_2(k,m) f(x_m)))^2 + \sum_{k=1}^{|S|} P_1(j,k)(\sum_{m=1}^{|S|} P_2(k,m) f(x_m))^2 \\
=& \sum_{k \in S, q \in S, k < q} P_1(j,k) P_1(j,q) (\sum_{m=1}^{|S|} P_2(k,m) f(x_m) - \sum_{m=1}^{|S|} P_2(q,m) f(x_m))^2 \\
=& \sum_{k,q,m,l \in S, k<q, m<l} \mathcal{B}(k,q,m,l) (f(x_l) - f(x_m))^2.
\end{aligned} \tag{4.17}$$

In the last equation of (4.17) we use notation $\mathcal{B}(k,q,m,l) := P_1(j,k)P_1(j,q)(P_2(k,m) - P_2(q,m))(P_2(q,l) - P_2(k,l))$.

For each $\mathcal{A}_{1,j}^{(\frac{\tau}{2})}$, we can simplify it to be

$$\begin{aligned}\mathcal{A}_{1,j}^{(\frac{\tau}{2})} =& -(\sum_{m=1}^{|S|} P_2(j,m)f(x_m))^2 + \sum_{m=1}^{|S|} P_2(j,m)f^2(x_m)\\ =& \sum_{m\in S, l\in S, m<l} P_2(j,m)P_2(j,l)(f(x_m)-f(x_l))^2.\end{aligned} \tag{4.18}$$

Combining (4.17) and (4.18), $V_{\frac{\tau}{2}}^{\text{op}}$ has the form

$$\begin{aligned}V_{\frac{\tau}{2}}^{\text{op}} =& \frac{1}{2}\left(\sum_{j=1,2,\ldots,|S|} \mu_0(x_j)\sqrt{\mathcal{A}_{0,j}^{(\frac{\tau}{2})}} + \sum_{k=1,2,\ldots,|S|} \mu_{\frac{\tau}{2}}(x_k)\sqrt{\mathcal{A}_{1,k}^{(\frac{\tau}{2})}}\right)^2\\ =& \frac{1}{2}\Bigg(\sum_{j=1,2,\ldots,|S|} \mu_0(x_j)\sqrt{\sum_{k,q,m,l\in S, k<q, m<l} \mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\ &+ \sum_{k=1}^{|S|}(\sum_{j=1}^{|S|}\mu_0(x_j)P_1(j,k))\sqrt{\sum_{m,l\in S, m<l} P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Bigg)^2\\ =& \frac{1}{2}\Bigg(\sum_{j=1}^{|S|}\mu_0(x_j)\Big(\sqrt{\sum_{k,q,m,l\in S, k<q, m<l} \mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\ &+ \sum_{k\in S} P_1(j,k)\sqrt{\sum_{m,l\in S, m<l} P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)\Bigg)^2\end{aligned} \tag{4.19}$$

According to Cauchy's Inequality,

$$\begin{aligned}V_{\frac{\tau}{2}}^{\text{op}} \le& \frac{|S|}{2}\sum_{j=1}^{|S|}\mu_0^2(x_j)\Big(\sqrt{\sum_{k,q,m,l\in S, k<q, m<l} \mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\ &+ \sum_{k\in S} P_1(j,k)\sqrt{\sum_{m,l\in S, m<l} P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2.\end{aligned} \tag{4.20}$$

Comparing (4.16) and (4.20), $V_{\frac{\tau}{2}}^{\text{op}} \le V_{\tau}^{\text{uni}}$ stands if for every $j \in S$,

$$\begin{aligned}&\Big(\sqrt{\sum_{k,q,m,l\in S, k<q, m<l} \mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\ &+ \sum_{k\in S} P_1(j,k)\sqrt{\sum_{m,l\in S, m<l} P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\\ \le& 2\sum_{m\in S, l\in S, m<l} R(j,m)R(j,l)(f(x_m)-f(x_l))^2.\end{aligned} \tag{4.21}$$

Therefore, in the remainder of this proof we prove (4.21) stands for arbitrary j.

Using Cauchy's Inequality,

$$\begin{aligned}&\Big(\sqrt{\sum_{k,q,m,l\in S,k<q,m<l}\mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\&+\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\\\leq&2\sum_{k,q,m,l\in S,k<q,m<l}\mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2\\&+2\Big(\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\end{aligned}\tag{4.22}$$

Using the definition of $\mathcal{B}(k,q,m,l)$, we have

$$\begin{aligned}&\sum_{k,q,m,l\in S,k<q,m<l}\mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2\\=&\sum_{k,q,m,l\in S,k<q,m<l}P_1(j,k)P_1(j,q)P_2(k,m)P_2(q,l)(f(x_l)-f(x_m))^2\\+&\sum_{k,q,m,l\in S,k<q,m<l}P_1(j,k)P_1(j,q)P_2(q,m)P_2(k,l)(f(x_l)-f(x_m))^2\\-&\sum_{k,q,m,l\in S,k<q,m<l}P_1(j,k)P_1(j,q)P_2(k,m)P_2(k,l)(f(x_l)-f(x_m))^2\\-&\sum_{k,q,m,l\in S,k<q,m<l}P_1(j,k)P_1(j,q)P_2(q,m)P_2(q,l)(f(x_l)-f(x_m))^2\end{aligned}\tag{4.23}$$

We expand $\Big(\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2$ by

$$\begin{aligned}&\Big(\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\\=&\sum_{k\in S}P_1^2(j,k)\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2+2\sum_{k,q\in S,k<q}P_1(j,k)P_1(j,q)\\&\sqrt{(\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2)(\sum_{m,l\in S,m<l}P_2(q,m)P_2(q,l)(f(x_m)-f(x_l))^2)}\\&.\end{aligned}\tag{4.24}$$

We apply (4.23) and (4.24) to (4.22), and use the fact that

$$\begin{aligned}
&\sum_{k,q,m,l\in S,k<q,m<l} P_1(j,k)P_1(j,q)P_2(k,m)P_2(q,l)(f(x_l)-f(x_m))^2\\
&+\sum_{k,q,m,l\in S,k<q,m<l} P_1(j,k)P_1(j,q)P_2(q,m)P_2(k,l)(f(x_l)-f(x_m))^2\\
&+\sum_{k\in S} P_1^2(j,k)\sum_{m,l\in S,m<l} P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2\\
&=\sum_{m\in S,l\in S,m<l}(\sum_{k\in S} P_1(j,k)P_2(k,m))(\sum_{q\in S} P_1(j,q)P_2(q,l))(f(x_m)-f(x_l))^2\\
&=\sum_{m\in S,l\in S,m<l} R(j,m)R(j,l)(f(x_m)-f(x_l))^2.
\end{aligned}\tag{4.25}$$

So the Inequality (4.22) is now

$$\begin{aligned}
&\Big(\sqrt{\sum_{k,q,m,l\in S,k<q,m<l}\mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2}\\
&\quad+\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\\
&\leq 2\sum_{k,q,m,l\in S,k<q,m<l}\mathcal{B}(k,q,m,l)(f(x_l)-f(x_m))^2\\
&\quad+2\Big(\sum_{k\in S}P_1(j,k)\sqrt{\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2}\Big)^2\\
&=2\sum_{m\in S,l\in S,m<l}R(j,m)R(j,l)(f(x_m)-f(x_l))^2\\
&\quad-2\sum_{k,q\in S,k<q}P_1(j,k)P_1(j,q)\Big(\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_l)-f(x_m))^2\\
&\quad+\sum_{m,l\in S,m<l}P_2(q,m)P_2(q,l)(f(x_l)-f(x_m))^2\\
&\quad-2\sqrt{(\sum_{m,l\in S,m<l}P_2(k,m)P_2(k,l)(f(x_m)-f(x_l))^2)(\sum_{m,l\in S,m<l}P_2(q,m)P_2(q,l)(f(x_m)-f(x_l))^2)}\Big)\\
&\leq 2\sum_{m\in S,l\in S,m<l}R(j,m)R(j,l)(f(x_m)-f(x_l))^2,
\end{aligned}\tag{4.26}$$

where the last inequality is an application of AM-GM inequality.

Since (4.21) stands for every j, $V^{\text{op}}_{\frac{\tau}{2}}\leq V^{\text{uni}}_{\tau}$ when $\kappa=1$.

□

Proof of Theorem 23

Proof of Theorem 23. We use mathematical induction to prove $V^{\mathrm{op}}_{\frac{\tau}{2}} \leq V^{\mathrm{uni}}_{\tau}$ for arbitrary positive integer κ. When $\kappa = 1$, $V^{\mathrm{op}}_{\frac{\tau}{2}} \leq V^{\mathrm{uni}}_{\tau}$ has been already proved in Lemma 2.

Suppose that when $\kappa = K$, $V^{\mathrm{op}}_{\frac{\tau}{2}} \leq V^{\mathrm{uni}}_{\tau}$. That is, the following inequality holds:

$$\begin{aligned} &\frac{1}{2}\left(\sum_{i=0,1,\dots,2K-1,j=1,2,\dots,|S|} \mu_{i\frac{\tau}{2}}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}\right)^2 \\ \leq &K|S|(\sum_{i=0,1,2,\dots,K-1,j=1,2,\dots,|S|} \mu^2_{i\tau}(x_j)\mathcal{A}^{(\tau)}_{i,j}). \end{aligned} \tag{4.27}$$

When $\kappa = K + 1$, we define

$$B_K := \sum_{i=0,1,\dots,2K-1,j=1,2,\dots,|S|} \mu_{i\frac{\tau}{2}}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}$$

and

$$C_K := \sum_{j=1,2,\dots,|S|} \mu_{K\tau}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{2K,j}} + \sum_{j=1,2,\dots,|S|} \mu_{(2K+1)\frac{\tau}{2}}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{2K+1,j}}$$

and have

$$\begin{aligned} V^{\mathrm{op}}_{\frac{\tau}{2}} =&\frac{1}{2}\left(\sum_{i=0,1,\dots,2K+1,j=1,2,\dots,|S|} \mu_{i\frac{\tau}{2}}(x_j)\sqrt{\mathcal{A}^{(\frac{\tau}{2})}_{i,j}}\right)^2 \\ =&\frac{1}{2}\left(B_K + C_K\right)^2 \\ \leq&\frac{1}{2}(\frac{1}{K}B_K^2 + C_K^2)(K+1), \end{aligned} \tag{4.28}$$

where the last inequality is an application of Cauchy–Schwarz inequality.

By letting $\tilde{f} := P^{(\tau)}_K f$ and applying (4.27) with $\tilde{f}$, we have

$$B_K^2 \leq 2K|S|(\sum_{i=0,1,2,\dots,K-1,j=1,2,\dots,|S|} \mu^2_{i\tau}(x_j)\mathcal{A}^{(\tau)}_{i,j}). \tag{4.29}$$

By applying Lemma 2 with $\tilde{\mu}_0 := \mu_0 \prod_{k=0}^{K-1} P^{(\tau)}_k$, we have

$$C_K^2 \leq 2|S|(\sum_{j=1,2,\dots,|S|} \mu^2_{K\tau}(x_j)\mathcal{A}^{(\tau)}_{K,j}). \tag{4.30}$$

Therefore,

$$\begin{aligned} V^{\mathrm{op}}_{\frac{\tau}{2}} \leq (K+1)|S|(&\sum_{i=0,1,2,\dots,K-1,j=1,2,\dots,|S|} \mu^2_{i\tau}(x_j)\mathcal{A}^{(\tau)}_{i,j} + \sum_{j=1,2,\dots,|S|} \mu^2_{K\tau}(x_j)\mathcal{A}^{(\tau)}_{K,j}) \\ = V^{\mathrm{uni}}_{\tau}&. \end{aligned} \tag{4.31}$$

□

4.C Omitted Proofs of Section 4.5

Proof of Proposition 24. We divide the proof of Proposition 24 into two parts. In the first part, we prove that

$$\sum_{j=1}^{|S|} \mu_{i\tau}(x_j)\mathcal{A}_{i,j}^{(\tau)} = C(T-(i+1)\tau) - C(T-i\tau), \tag{4.32}$$

where the definition of $\mu_{i\tau}(x_j)$ and $\mathcal{A}_{i,j}^{(\tau)}$ are given in (4.1) and (4.2). In the second part, we combine (4.32) with Theorem 19 to obtain the scale of asymptotic variance for $\hat{h}_T^{N,\kappa}$.

Denote $b_i := (\mathbb{1}_{\{X_{i\tau}=x_1\}}, \mathbb{1}_{\{X_{i\tau}=x_2\}}, \dots, \mathbb{1}_{\{X_{i\tau}=x_{|S|}\}})^\intercal \in \mathcal{R}^{|S|}$, then

$$\mathbb{E}(f(X_T)|X_{i\tau}) = b_i \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f,$$

and

$$\mathrm{Var}(\mathbb{E}(f(X_T)|X_{i\tau})) = f^\intercal(\prod_{k=i}^{\kappa-1} P_k^{(\tau)})^\intercal \mathrm{Cov}(b_i) \prod_{k=i}^{\kappa-1} P_k^{(\tau)} f.$$

For $k, m \in S$, the (k,m) entry of $\mathrm{Cov}(b_i)$ is $-\mu_{i\tau}(x_k)\mu_{i\tau}(x_m)$ if $k \neq m$, and is $\mu_{i\tau}(x_k)(1 - \mu_{i\tau}(x_k))$ if $k = m$.

Similarly we have

$$\mathrm{Var}(\mathbb{E}(f(X_T)|X_{(i+1)\tau})) = f^\intercal(\prod_{k=i+1}^{\kappa-1} P_k^{(\tau)})^\intercal \mathrm{Cov}(b_{i+1}) \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f.$$

Here we set $\prod_{k=i+1}^{\kappa-1} P_k^{(\tau)}$ to equal the identity matrix I if $i = \kappa - 1$. So

$$C(T-(i+1)\tau) - C(T-i\tau) = f^\intercal(\prod_{k=i+1}^{\kappa-1} P_k^{(\tau)})^\intercal (\mathrm{Cov}(b_{i+1}) - (P_i^{(\tau)})^\intercal \mathrm{Cov}(b_i) P_i^{(\tau)}) \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f. \tag{4.33}$$

On the other hand,

$$\begin{aligned}&\sum_{j=1}^{|S|} \mu_{i\tau}(x_j)\mathcal{A}_{i,j}^{(\tau)} \\ =&f^\intercal(\prod_{k=i+1}^{\kappa-1} P_k^{(\tau)})^\intercal \sum_{j=1}^{|S|} \mu_{i\tau}(x_j)(\mathrm{diag}\{P_i^\tau(j,x_1), P_i^\tau(j,x_2), \dots, P_i^\tau(j,x_{|S|})\} - v_{ij}v_{ij}^\intercal) \prod_{k=i+1}^{\kappa-1} P_k^{(\tau)} f.\end{aligned}$$

Here $v_{ij} = (P_i^\tau(j,x_1), P_i^\tau(j,x_2), \dots, P_i^\tau(j,x_{|S|}))^\intercal$.

Denote $M_i := \sum_{j=1}^{|S|} \mu_{i\tau}(x_j)(\mathrm{diag}\{P_i^\tau(j,x_1), P_i^\tau(j,x_2), \ldots, P_i^\tau(j,x_{|S|})\} - v_{ij}v_{ij}^\intercal)$. For $k,m \in S$, the (k,m) entry of M_i is

$$-\sum_{j=1}^{|S|} \mu_{i\tau}(x_j) P_i^\tau(j,x_k) P_i^\tau(j,x_m)$$

if $k \neq m$, and

$$\sum_{j=1}^{|S|} \mu_{i\tau}(x_j) P_i^\tau(j,x_k)(1 - P_i^\tau(j,x_k))$$

if $k = m$. It can be verified that $M_i = \mathrm{Cov}(b_{i+1}) - (P_i^{(\tau)})^\intercal \mathrm{Cov}(b_i) P_i^{(\tau)}$. Therefore, (4.32) is proved.

According to Theorem 19,

$$\sqrt{N}(\hat{h}_T^{N,\kappa} - \mathbb{E}\, f(X_T)) \xrightarrow{d} \mathcal{N}(0, \sum_{i=0,1,2,\ldots,\kappa-1, j=1,2,\ldots,|S|} \frac{\mu_i^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}}).$$

So the asymptotic variance is $\frac{1}{N}\sum_{i=0,1,2,\ldots,\kappa-1,j=1,2,\ldots,|S|} \frac{\mu_i^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}}$. Define $C' := \sum_{k=1}^{\infty} k^{\frac{\gamma-1}{2}}$. We have

$$\begin{aligned}
&\frac{1}{N} \sum_{i=0,1,2,\ldots,\kappa-1,j=1,2,\ldots,|S|} \frac{\mu_i^2(x_j)\mathcal{A}_{i,j}^{(\tau)}}{p_{i,j}} \\
\leq &\frac{1}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{(\sum_{j=1}^{|S|} \mu_{i\tau}(x_j))(\sum_{j=1}^{|S|} \mu_{i\tau}(x_j)\mathcal{A}_{i,j}^{(\tau)})}{p_{i,j}} \\
= &\frac{1}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{C(T-(i+1)\tau) - C(T-i\tau)}{p_{i,j}} \\
\leq &\frac{C'|S|}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{C(T-(i+1)\tau) - C(T-i\tau)}{(\kappa-i)^{\frac{\gamma-1}{2}}} \\
= &\frac{C'|S|}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{C(i\tau) - C((i+1)\tau)}{(i+1)^{\frac{\gamma-1}{2}}}
\end{aligned} \tag{4.34}$$

For $C(i\tau) - C((i+1)\tau)$ we apply the inequality $x - y \leq 2\sqrt{x}(\sqrt{x} - \sqrt{y})$, which holds for

$x \geq 0$ and $y \geq 0$. We have

$$\begin{aligned}
&\frac{C'|S|}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{C(i\tau) - C((i+1)\tau)}{(i+1)^{\frac{\gamma-1}{2}}} \\
\leq &\frac{C'|S|}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \frac{2\sqrt{C(i\tau)}(\sqrt{C(i\tau)} - \sqrt{C((i+1)\tau)})}{(i+1)^{\frac{\gamma-1}{2}}} \\
\leq &\frac{2|S|\sqrt{c'}C'\tau^{\frac{\gamma}{2}}}{N} \sum_{i=0,1,2,\ldots,\kappa-1} \sqrt{i+1}(\sqrt{C(i\tau)} - \sqrt{C((i+1)\tau)}) \\
\leq &\frac{2|S|\sqrt{c'}C'\tau^{\frac{\gamma}{2}}}{N} \sum_{i=0,1,2,\ldots,\kappa-1} (\sqrt{i+1} - \sqrt{i})\sqrt{C(i\tau)}
\end{aligned}$$

As $\sqrt{i+1} - \sqrt{i} \leq (i+1)^{-\frac{1}{2}}$ for $i \geq 0$, we have

$$\begin{aligned}
&\frac{2|S|\sqrt{c'}C'\tau^{\frac{\gamma}{2}}}{N} \sum_{i=0,1,2,\ldots,\kappa-1} (\sqrt{i+1} - \sqrt{i})\sqrt{C(i\tau)} \\
\leq &\frac{2|S|c'C'\tau^{\frac{\gamma}{2}}}{N} \sum_{i=0,1,2,\ldots,\kappa-1} (i+1)^{\frac{\gamma-1}{2}} \\
\leq &\frac{2|S|c'C'\tau^{\frac{\gamma}{2}}}{N} \sum_{i=0}^{\infty} (i+1)^{\frac{\gamma-1}{2}} \\
\leq &\frac{2|S|c'(C')^2\tau^{\frac{\gamma}{2}}}{N}.
\end{aligned}$$

Therefore Proposition 24 is proved. □

Chapter 5

Conclusions

This thesis discusses several challenges occurring in the process of simulation-based decision making. In our study, we take into consideration the management of computational resources, and investigate on the trade-off between the accuracy of decision making and resource saving. We explore on several directions to achieve efficient simulation algorithm design: exploiting the correlation structure of different stochastic systems and balancing input uncertainty and simulation error; making use of approximating systems; improving the simulation techniques and adapting better parallel simulation policies. We briefly list our contributions and future directions as follows.

In Chapter 2, we consider the target of selecting the system with better expected performance between two stochastic systems, given fixed resource budget. The resource can be used to collect more input data to reduce input uncertainty, and to implement more simulation replications to eliminate simulation error. We model the resource allocation problem as an optimization problem. The objective is appropriately allocating resource to maximize the probability of correctly selecting the better system. We exploit correlation structure of input data and common random numbers in simulation to save costs, and provide closed-form optimal resource allocation solutions. Future work may include extensions to comparison of a larger number of different systems, and numerical experiments based on real data and examples.

In Chapter 3, we present a framework to use a sequence of approximating systems to optimize a stochastic system that has complicated stochastic structure and cannot be simulated exactly with finite computational cost. With this framework, we propose new gradient-based simulation-optimization algorithms that utilize the approximations with increasing resolution and higher simulation cost to construct stochastic gradients and perform gradient search. To circumvent the challenge that the objective functions associated with the approximating systems are discontinuous, we use the finite difference method to construct gradient estimators for approximating systems. Under the assumption that the objective function of the original system is strongly convex and smooth, we prove algorithm convergence, convergence rate, and optimality of algorithm design, without any assumption on convexity or continuity with the approximating systems. We demonstrate the dependence on the dimension of the decision variable for the algorithm performance and optimal parameters choices. We then present a multilevel version of the proposed algorithms to further improve convergence rates, when in addition the sequence of approximations can be naturally coupled. We prove theoretically and then show empirically that the additional use of multilevel structure can further improve the computational efficiency of the proposed simulation-optimization algorithms.

In Chapter 4, we consider simulation problem in presence of limited time slot, and propose a framework for time-parallel simulation for a time-varying Markov chain with finite state space. We provide a close-form optimal processor assignment policy through a central limit theorem. Further, we propose a two-stage simulation procedure that learns the near optimal policy adaptively and show this procedure can reduce the mean square error theoretically and empirically. Future work may include extensions to Markov chain models with infinite state space.

Bibliography

[1] Ahmad Ajalloeian and Sebastian U. Stich. *On the Convergence of SGD with Biased Gradients*. 2020. DOI: 10.48550/ARXIV.2008.00051. URL: https://arxiv.org/abs/2008.00051.

[2] Søren Asmussen and Peter W Glynn. *Stochastic simulation: algorithms and analysis*. Vol. 57. Springer, 2007.

[3] Vladmir Balabanov and Gerhard Venter. "Multi-fidelity optimization with high-fidelity analysis and low-fidelity gradients". In: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2014, p. 4459.

[4] Jerry Banks. *Discrete event system simulation*. Pearson Education India, 2005.

[5] Russell R. Barton, Barry L. Nelson, and Wei Xie. "Quantifying Input Uncertainty via Simulation Confidence Intervals". In: *INFORMS Journal on Computing* 26.1 (2014), pp. 74–87. DOI: 10.1287/ijoc.2013.0548. eprint: https://doi.org/10.1287/ijoc.2013.0548. URL: https://doi.org/10.1287/ijoc.2013.0548.

[6] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear Programming: Theory and Algorithms*. Hoboken, New Jersey: John Wiley & Sons, 2013.

[7] Jean Bélanger, Philippe Venne, Jean-Nicolas Paquin, et al. "The what, where and why of real-time simulation". In: *Planet Rt* 1.1 (2010), pp. 25–29.

[8] Jose H Blanchet, Peter W Glynn, and Yanan Pei. "Unbiased multilevel Monte Carlo: Stochastic optimization, steady-state simulation, quantiles, and other applications". In: *arXiv preprint arXiv:1904.09929* (2019).

[9] K Mani Chandy. "Space, time, and simulation". In: *Proceedings of the SCS Multiconference on Distributed Simulation*. Vol. 21. 2. 1989, pp. 53–57.

[10] Chun-hung Chen and Loo Hay Lee. *Stochastic Simulation Optimization: an Optimal Computing Budget Allocation*. Vol. 1. World scientific, 2011.

[11] Chun-Hung Chen et al. "Ranking and Selection: Efficient Simulation Budget Allocation". In: *Handbook of Simulation Optimization*. Ed. by M.C. Fu. New York: Springer-Verlag, 2015, pp. 45–80.

[12] Russell CH Cheng and Wayne Holloand. “Sensitivity of Computer Simulation Experiments to Errors in Input Data”. In: *Journal of Statistical Computation and Simulation* 57.1-4 (1997), pp. 219–241.

[13] Stephen E Chick. “Input Distribution Selection for Simulation Experiments: Accounting for Input Uncertainty”. In: *Operations Research* 49.5 (2001), pp. 744–758.

[14] Linsen Chong and Carolina Osorio. “A simulation-based optimization algorithm for dynamic large-scale urban transportation problems”. In: *Transportation Science* 52.3 (2018), pp. 637–656.

[15] K. Cooper, S. R. Hunter, and K. Nagaraj. “Bi-objective simulation optimization on integer lattices using the epsilon-constraint method in a retrospective approximation framework”. In: *INFORMS Journal on Computing* 32.4 (2020), pp. 1080–1100. DOI: `10.1287/ijoc.2019.0918`.

[16] Canan G Corlu and Bahar Biller. “A Subset Selection Procedure under Input Parameter Uncertainty”. In: *Proceedings of the 2013 Winter Simulation Conference*. Ed. by R. Pasupathy et al. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey, 2013, pp. 463–473.

[17] L Dai and CH Chen. “Rates of Convergence of Ordinal Comparison for Dependent Discrete Event Dynamic Systems”. In: *Journal of Optimization Theory and Applications* 94.1 (1997), pp. 29–54.

[18] Thu Ha Dao Thi, Jean-Michel Fourneau, and Franck Quessette. “Time-parallel simulation for stochastic automata networks and stochastic process algebra”. In: *Analytical and Stochastic Modeling Techniques and Applications: 21st International Conference, ASMTA 2014, Budapest, Hungary, June 30–July 2, 2014. Proceedings 21*. Springer. 2014, pp. 140–154.

[19] Vanda De Angelis, Giovanni Felici, and Paolo Impelluso. “Integrating simulation and optimisation in health care centre management”. In: *European Journal of Operational Research* 150.1 (2003), pp. 101–114.

[20] Steffen Dereich. “General multilevel adaptations for stochastic approximation algorithms II: CLTs”. In: *Stochastic Processes and their Applications* (2020).

[21] Steffen Dereich and Thomas Müller-Gronbach. “General multilevel adaptations for stochastic approximation algorithms of Robbins–Monro and Polyak–Ruppert type”. In: *Numerische Mathematik* 142.2 (2019), pp. 279–328.

[22] John C Duchi et al. “Optimal rates for zero-order convex optimization: The power of two function evaluations”. In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2788–2806.

[23] John C. Duchi et al. “Optimal Rates for Zero-Order Convex Optimization: The Power of Two Function Evaluations”. In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2788–2806.

[24] David J. Eckman and Shane G. Henderson. "Biased gradient estimators in simulation optimization". In: *Proceedings of the 2020 Winter Simulation Conference*. Ed. by K.-H. Bae et al. IEEE. Piscataway NJ, 2020, Submitted.

[25] Vaclav Fabian et al. "On asymptotic normality in stochastic approximation". In: *The Annals of Mathematical Statistics* 39.4 (1968), pp. 1327–1332.

[26] Alois Ferscha and Satish K Tripathi. *Parallel and distributed simulation of discrete event systems*. Tech. rep. 1998.

[27] Noufel Frikha et al. "Multi-level stochastic approximation algorithms". In: *The Annals of Applied Probability* 26.2 (2016), pp. 933–985.

[28] Michael C Fu et al. *Handbook of simulation optimization*. Vol. 216. Springer, 2015.

[29] Michael C Fu and Jian-Qiang Hu. *Conditional Monte Carlo: Gradient estimation and optimization applications*. Vol. 392. Springer Science & Business Media, 2012.

[30] Michael C Fu et al. "Simulation Allocation for Determining the Best Design in the Presence of Correlated Sampling". In: *INFORMS Journal on Computing* 19.1 (2007), pp. 101–111.

[31] Richard M Fujimoto. "Parallel discrete event simulation". In: *Communications of the ACM* 33.10 (1990), pp. 30–53.

[32] Richard M Fujimoto. "Parallel discrete event simulation: Will the field survive?" In: *ORSA Journal on Computing* 5.3 (1993), pp. 213–230.

[33] Richard M Fujimoto, Ioanis Nikolaidis, and C Anthony Cooper. "Parallel simulation of statistical multiplexers". In: *Discrete Event Dynamic Systems* 5 (1995), pp. 115–140.

[34] Richard M Fujimoto et al. "Parallel discrete event simulation: The making of a field". In: *2017 Winter Simulation Conference (WSC)*. IEEE. 2017, pp. 262–291.

[35] Saeed Ghadimi and Guanghui Lan. "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2341–2368.

[36] Michael B Giles. "Multilevel monte carlo path simulation". In: *Operations research* 56.3 (2008), pp. 607–617.

[37] Paul Glasserman and Pirooz Vakili. "Comparing Markov Chains Simulated in Parallel". In: *Probability in the Engineering and Informational Sciences* 8.3 (1994), pp. 309–326.

[38] Peter Glynn and Sandeep Juneja. "Selecting the Best System and Multi-armed Bandits". In: *arXiv preprint arXiv:1507.04564* (2015). `https://arxiv.org/abs/1507.04564`, accessed 16th February 2020.

[39] Peter W Glynn. "Likelihood ratio gradient estimation for stochastic systems". In: *Communications of the ACM* 33.10 (1990), pp. 75–84.

[40] Wei-Bo Gong and Yu-Chi Ho. “Smoothed (conditional) perturbation analysis of discrete event dynamical systems”. In: *IEEE Transactions on Automatic Control* 32.10 (1987), pp. 858–866.

[41] Lars Grasedyck et al. “Space and time parallel multigrid for optimization and uncertainty quantification in PDE simulations”. In: *Software for Exascale Computing-SPPEXA 2013-2015*. Springer. 2016, pp. 507–523.

[42] Albert G Greenberg, Boris D Lubachevsky, and Isi Mitrani. “Algorithms for unboundedly parallel simulations”. In: *ACM Transactions on Computer Systems (TOCS)* 9.3 (1991), pp. 201–221.

[43] Andreas Griewank et al. “On automatic differentiation”. In: *Mathematical Programming: recent developments and applications* 6.6 (1989), pp. 83–107.

[44] A Haurie, P L’Ecuyer, and Ch Van Delft. “Monte-Carlo optimization of parametrized policies in a class of piecewise deterministic control systems arising in manufacturing flow control”. In: *Discrete Event Dynamic Systems: Theory and Applications* 4.1 (1994), pp. 87–111.

[45] Philip Heidelberger and Harold S Stone. *Parallel trace-driven cache simulation by time partitioning*. Tech. rep. Institute of Electrical and Electronics Engineers (IEEE), 1990.

[46] Bernd Heidergott and FJ Vázquez-Abad. “Measure-valued differentiation for Markov chains”. In: *Journal of Optimization Theory and Applications* 136.2 (2008), pp. 187–209.

[47] Bernd Heidergott et al. “Gradient estimation for discrete-event systems by measure-valued differentiation”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20.1 (2010), pp. 1–28.

[48] Stefan Heinrich. “Multilevel monte carlo methods”. In: *International Conference on Large-Scale Scientific Computing*. Springer. 2001, pp. 58–67.

[49] Desmond J Higham. “An algorithmic introduction to numerical simulation of stochastic differential equations”. In: *SIAM review* 43.3 (2001), pp. 525–546.

[50] Stacy D Hill and Michael C Fu. “Simulation optimization via simultaneous perturbation stochastic approximation”. In: *Proceedings of Winter Simulation Conference*. IEEE. 1994, pp. 1461–1464.

[51] L Jeff Hong, Weiwei Fan, and Jun Luo. “Review on ranking and selection: A new perspective”. In: *Frontiers of Engineering Management* 8.3 (2021), pp. 321–343.

[52] L Jeff Hong, Yingda Song, and Tan Wang. “Fast Discrete-Event Simulation of Markovian Queueing Networks through Euler Approximation”. In: *arXiv:2402.13259* (2024).

[53] L Jeff Hong and Xiaowei Zhang. “Surrogate-Based Simulation Optimization”. In: *arXiv preprint arXiv:2105.03893* (2021).

[54] Yifan Hu et al. “Biased Stochastic Gradient Descent for Conditional Stochastic Optimization”. In: *arXiv preprint arXiv:2002.10790* (2020).

[55] De Huang et al. *Matrix Concentration for Products*. 2020. DOI: `10.48550/ARXIV.2003.05437`. URL: `https://arxiv.org/abs/2003.05437`.

[56] Deng Huang et al. “Sequential kriging optimization using multiple-fidelity evaluations”. In: *Structural and Multidisciplinary Optimization* 32.5 (2006), pp. 369–382.

[57] Susan R Hunter et al. “An introduction to multiobjective simulation optimization”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 29.1 (2019), pp. 1–36.

[58] Jean Jacod, Philip Protter, et al. “Asymptotic error distributions for the Euler method for stochastic differential equations”. In: *The Annals of Probability* 26.1 (1998), pp. 267–307.

[59] Nanjing Jian and Shane G Henderson. “An introduction to simulation optimization”. In: *2015 Winter Simulation Conference (wsc)*. IEEE. 2015, pp. 1780–1794.

[60] Nabil Kahalé. “On the effective dimension and multilevel Monte Carlo”. In: *Operations Research Letters* 50.4 (2022), pp. 415–421.

[61] Zohar Karnin, Tomer Koren, and Oren Somekh. “Almost Optimal Exploration in Multi-armed Bandits”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Stroudsburg, Pennsylvania: IMLS, 2013, pp. 1238–1246.

[62] Emilie Kaufmann and Shivaram Kalyanakrishnan. “Information Complexity in Bandit Subset Selection”. In: *Proceedings of Machine Learning Research*. 2013, pp. 228–251.

[63] Seksan Kiatsupaibul, Robert L Smith, and Zelda B Zabinsky. “Single observation adaptive search for continuous simulation optimization”. In: *Operations research* 66.6 (2018), pp. 1713–1727.

[64] Tobias Kiesling and Siegfried Pohl. “Time-parallel simulation with approximative state matching”. In: *Proceedings of the eighteenth workshop on Parallel and distributed simulation*. 2004, pp. 195–202.

[65] Seong-Hee Kim and B. L. Nelson. “Recent Advances in Ranking and Selection”. In: *Proceedings of the 2007 Winter Simulation Conference*. Ed. by S. G. Henderson et al. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey, 2007, pp. 162–172.

[66] Seong-Hee Kim and Barry L Nelson. “A Fully Sequential Procedure for Indifference-Zone Selection in Simulation”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 11.3 (2001), pp. 251–273.

[67] Young Jin Kim, Dimitri Mavris, and Richard Fujimoto. “Time-and space-parallel simulation of air traffic networks”. In: *Simulation* 95.12 (2019), pp. 1213–1228.

[68] Nathan L Kleinman, James C Spall, and Daniel Q Naiman. "Simulation-based optimization with stochastic approximation using common random numbers". In: *Management Science* 45.11 (1999), pp. 1570–1578.

[69] Harold Joseph Kushner and Dean S Clark. *Stochastic approximation methods for constrained and unconstrained systems.* Vol. 26. Springer Science & Business Media, 2012.

[70] Pierre L'ecuyer. "A unified view of the IPA, SF, and LR gradient estimation techniques". In: *Management Science* 36.11 (1990), pp. 1364–1383.

[71] Pierre L'Ecuyer and George Yin. "Budget-dependent convergence rate of stochastic approximation". In: *SIAM Journal on Optimization* 8.1 (1998), pp. 217–247.

[72] Averill M Law, W David Kelton, and W David Kelton. *Simulation modeling and analysis.* Vol. 3. Mcgraw-hill New York, 2007.

[73] Yi-Bing Lin and Edward D Lazowska. "A time-division algorithm for parallel simulation". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 1.1 (1991), pp. 73–83.

[74] Jason Liu, James J Cochran, and Louis A Cox. "Parallel discrete-event simulation". In: *Wiley Online Library* 35 (2009), p. 101.

[75] Lennart Ljung. "Analysis of recursive stochastic algorithms". In: *IEEE transactions on automatic control* 22.4 (1977), pp. 551–575.

[76] Jun Luo et al. "Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments". In: *Operations Research* 63.5 (2015), 1177–1194.

[77] Barry L Nelson and Frank J Matejcik. "Using Common Random Numbers for Indifference-Zone Selection and Multiple Comparisons in Simulation". In: *Management Science* 41.12 (1995), pp. 1935–1945.

[78] Yurii Nesterov and Vladimir Spokoiny. "Random gradient-free minimization of convex functions". In: *Foundations of Computational Mathematics* 17.2 (2017), pp. 527–566.

[79] Eric C. Ni et al. "Comparing Message Passing Interface and MapReduce for large-scale parallel ranking and selection". In: *2015 Winter Simulation Conference (WSC).* 2015, pp. 3858–3867. DOI: 10.1109/WSC.2015.7408542.

[80] Raghu Pasupathy and Soumyadip Ghosh. "Simulation optimization: A concise overview and implementation guide". In: *Theory Driven by Influential Applications* (2013), pp. 122–150.

[81] Linda Pei, Barry L Nelson, and Susan Hunter. "A new framework for parallel ranking & selection using an adaptive standard". In: *2018 winter simulation conference (wsc).* IEEE. 2018, pp. 2201–2212.

[82] Linda Pei, Barry L Nelson, and Susan R Hunter. "Evaluation of bi-PASS for parallel simulation optimization". In: *2020 Winter Simulation Conference (WSC)*. IEEE. 2020, pp. 2960–2971.

[83] Yijie Peng et al. "A new unbiased stochastic derivative estimator for discontinuous sample performances with structural parameters". In: *Operations Research* 66.2 (2018), pp. 487–499.

[84] Dhananjai Madhava Rao et al. "Unsynchronized parallel discrete event simulation". In: *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*. Vol. 2. IEEE. 1998, pp. 1563–1570.

[85] Martin I Reiman and Alan Weiss. "Sensitivity analysis for simulations via likelihood ratios". In: *Operations Research* 37.5 (1989), pp. 830–844.

[86] Daniel Russo. "Simple Bayesian Algorithms for Best Arm Identification". In: *Operations Research* (2020). To Appear.

[87] Ilya O Ryzhov. "On the Convergence Rates of Expected Improvement Methods". In: *Operations Research* 64.6 (2016), pp. 1515–1528.

[88] Jay D Schwartz, Wenlin Wang, and Daniel E Rivera. "Simulation-based optimization of process control policies for inventory management in supply chains". In: *Automatica* 42.8 (2006), pp. 1311–1320.

[89] Prateek Sharma. "Discrete-event simulation". In: *Int. J. Sci. Technol. Res* 4.04 (2015), pp. 136–140.

[90] Eunhye Song and Barry L Nelson. "Input–output Uncertainty Comparisons for Discrete Optimization via Simulation". In: *Operations Research* 67.2 (2019), pp. 562–576.

[91] Eunhye Song, Barry L Nelson, and L Jeff Hong. "Input Uncertainty and Indifference-Zone Ranking & Selection". In: *Proceedings of the 2015 Winter Simulation Conference*. Ed. by L. Yilmaz et al. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey, 2015, pp. 414–424.

[92] Jie Song et al. "Multi-fidelity sampling for efficient simulation-based decision making in manufacturing management". In: *IISE Transactions* 51.7 (2019), pp. 792–805.

[93] James C Spall. "Adaptive stochastic approximation by the simultaneous perturbation method". In: *IEEE transactions on automatic control* 45.10 (2000), pp. 1839–1853.

[94] James C Spall et al. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation". In: *IEEE transactions on automatic control* 37.3 (1992), pp. 332–341.

[95] Sunil Thulasidasan and Stephan Eidenbenz. "Accelerating traffic microsimulations: A parallel discrete-event queue-based approach for speed and scale". In: *Proceedings of the 2009 Winter Simulation Conference (WSC)*. IEEE. 2009, pp. 2457–2466.

[96] Daniel Vasquez, Sara Shashaani, and Raghu Pasupathy. "Astro for derivative-based stochastic optimization: Algorithm description & numerical experiments". In: *2019 Winter Simulation Conference (WSC)*. IEEE. 2019, pp. 3563–3574.

[97] Jain J Wang and Marc Abrams. "Approximate time-parallel simulation of queueing systems with losses". In: *Proceedings of the 24th conference on Winter simulation*. 1992, pp. 700–708.

[98] Songhao Wang, Szu Hui Ng, and William Benjamin Haskell. "A Multilevel Simulation Optimization Approach for Quantile Functions". In: *INFORMS Journal on Computing* (2021).

[99] Tan Wang and L Jeff Hong. "Large-Scale Inventory Optimization: A Recurrent Neural Networks–Inspired Simulation Approach". In: *INFORMS Journal on Computing* 35.1 (2023), pp. 196–215.

[100] Tan Wang, Yingda Song, and Jeff Hong. "Fast Approximation to Discrete-Event Simulation of Markovian Queueing Networks". In: *2023 Winter Simulation Conference (WSC)*. IEEE. 2023, pp. 3613–3623.

[101] Di Wu and Enlu Zhou. "Analyzing and Provably Improving Fixed Budget Ranking and Selection Algorithms". In: *arXiv preprint arXiv:1811.12183* (2018). `https://arxiv.org/abs/1811.12183`, accessed 24th February 2020.

[102] Di Wu and Enlu Zhou. "Ranking and Selection under Input Uncertainty: Fixed Confidence and Fixed Budget". In: *arXiv preprint arXiv:1708.08526* (2017). `https://arxiv.org/abs/1708.08526`, accessed 14th February 2020.

[103] Hao Wu, Richard M Fujimoto, and Mostafa Ammar. "Time-parallel trace-driven simulation of CSMA/CD". In: *Seventeenth Workshop on Parallel and Distributed Simulation, 2003.(PADS 2003). Proceedings*. IEEE. 2003, pp. 105–112.

[104] Wei Xie, Barry L Nelson, and Russell R Barton. "A Bayesian Framework for Quantifying Uncertainty in Stochastic Simulation". In: *Operations Research* 62.6 (2014), pp. 1439–1452.

[105] Jie Xu et al. "Efficient multi-fidelity simulation optimization". In: *Proceedings of the 2014 Winter Simulation Conference*. 2014, pp. 3940–3951.

[106] Jie Xu et al. "Simulation optimization in the era of Industrial 4.0 and the Industrial Internet". In: *Journal of Simulation* 10.4 (2016), pp. 310–320.

[107] Jingxu Xu and Zeyu Zheng. "Online Supplement and Code Depository for Gradient-based Simulation Optimization Algorithms via Multi-Resolution System Approximations". In: (2023). https://github.com/INFORMSJoC/2021.0289. DOI: `105281/zenodo.7485443`.

[108] Anlu Yan and Wei-Bo Gong. "Time-driven fluid simulation for high-speed networks". In: *IEEE Transactions on Information Theory* 45.5 (1999), pp. 1588–1599.

[109] Wei-Ning Yang and Barry L Nelson. "Using Common Random Numbers and Control Variates in Multiple-comparison Procedures". In: *Operations Research* 39.4 (1991), pp. 583–591.

[110] G Yin, H Yan, and SXC Lou. "On a class of stochastic optimization algorithms with applications to manufacturing models". In: *Model-Oriented Data Analysis* (1993), pp. 213–226.

[111] Ilsoo Yun and Byungkyu "brian" Park. "Application of Stochastic Optimization Method for an Urban Corridor". In: *Proceedings of the 2006 Winter Simulation Conference*. 2006, pp. 1493–1499. DOI: `10.1109/WSC.2006.322918`.

[112] Haixiang Zhang, Zeyu Zheng, and Javad Lavaei. "Gradient-Based Algorithms for Convex Discrete Optimization via Simulation". In: *Operations Research* (2022).

[113] Haixiang Zhang, Zeyu Zheng, and Javad Lavaei. "Stochastic L-convex Function Minimization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 13004–13018.

[114] Ying Zhong et al. "Speeding up Paulson's procedure for large-scale problems using parallel computing". In: *INFORMS Journal on Computing* 34.1 (2022), pp. 586–606.

[115] Enlu Zhou and Shalabh Bhatnagar. "Gradient-based adaptive stochastic search for simulation optimization over continuous space". In: *INFORMS Journal on Computing* 30.1 (2017), pp. 154–167.