2013-04-20

Simulation-based Optimization over Discrete Sets with Noisy Constraints

Yao Luo
University of Miami, ly1987510@gmail.com

UNIVERSITY OF MIAMI

SIMULATION-BASED OPTIMIZATION OVER DISCRETE SETS WITH NOISY CONSTRAINTS

By

Yao Luo

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2013

UNIVERSITY OF MIAMI

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

SIMULATION-BASED OPTIMIZATION OVER DISCRETE SETS WITH NOISY CONSTRAINTS

Yao Luo

Approved:

_____ Eunji Lim, Ph.D. Assistant Professor of Industrial Engineering	_____ M. Brian Blake, Ph.D. Dean of the Graduate School
_____ Shihab S. Asfour, Ph.D. Professor of Industrial Engineering	_____ Nurcin Celik, Ph.D. Assistant Professor of Industrial Engineering
_____ Murat Erkoc, Ph.D. Associate Professor of Industrial Engineering	_____ Edward Baker, Ph.D. Professor of Management Science

LUO, YAO (Ph.D., Industrial Engineering)
Simulation-based Optimization over Discrete Sets with Noisy Constraints (May 2013)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Eunji Lim.
No. of pages in text. (166)

The first part of this dissertation studies a constrained simulation-based optimization problem over a discrete set where noise-corrupted observations of the objective and constraints are available. The problem is challenging because the feasibility of a solution cannot be known for certain. The uncertainty, in turn, arises from the noisy measurements of the constraints. To tackle this issue, we propose an innovative method that converts constrained optimization into the unconstrained optimization problem of finding a saddle point of the Lagrangian. The method applies stochastic approximation to the Lagrangian in search of the saddle point. We prove that the proposed method converges to the optimal solution almost surely (a.s.) under suitable conditions as the number of iterations grows. We present the effectiveness of the proposed method numerically in four examples, with applications in inventory control, call center staffing and emergency room management. The second part of this dissertation discusses the problem of fitting a convex function based on noisy data from simulation output. The traditional way of fitting a convex function to data, which is done by computing a convex function minimizing the sum of least squares, takes too long to compute the fit. It also runs into an "out of memory" issue when the number of data points exceeds a few hundred. In this

dissertation, we propose a computationally efficient way by minimizing the sum of least absolute deviations rather than the sum of squares. The least absolute deviations estimator we introduce in this dissertation is posed via a solution to a linear program (LP) while the traditional least squares estimator is posed via a solution to a quadratic program (QP). Furthermore, our LP formulation has a dual problem that exhibits a block-angular form in its constraints. This enables one to apply decomposition techniques such as Dantzig-Wolfe decomposition to solve the dual problem. Thus the proposed estimator can be computed faster and for larger datasets than the least squares estimator. We present numerical examples to illustrate the relative performance of the proposed estimator compared to that of the least squares estimator. We also establish the consistency of the proposed estimator and its derivative by proving that, under modest assumptions, the estimator and its derivative converge almost surely (a.s.) to the true values as the number of data points increases to infinity. The third part of this dissertation concerns the initial transient problems when running discrete-event simulation in our simulation-based optimization framework. When using a discrete event simulation to estimate some steady-state variables we are aiming to optimize, we deduce that it is desirable to initialize the simulation according to steady-state distribution because the initial transient phase will be induced otherwise. However, due to the lack of information on steady-state distribution, practitioners usually start the simulation in some arbitrary fashion, which results in an initial transient phase prior to steady-state. In this paper, we provide a methodology to determine the length of the initial transient phase of (possibly multi-dimensional) simulation output. Such an elaborated method can be further used to devise an algorithm to compute better estimators for steady-state performance measures

by utilizing simulation output after the initial transient phase. The proposed methodology is based on a simple idea of dividing simulation output into several batches, observing the way the observations are distributed in each batch, and trying to find a change in these distributions. The efficiency of the proposed methodology is illustrated through numerical experiments.

Acknowledgement

I would like to express my heartfelt gratitude and respect to my dissertation advisor, Dr. Eunji Lim, for her guidance and support throughout my pursuit of this research. The completion of this dissertation would not have been possible without her encouragement and help.

I would like to extend my sincerest thanks to Dr. Shihab S. Asfour, Dr. Murat Erkoc, Dr. Nurcin Celik, and Dr. Edward Baker, for serving on my dissertation committee. Their thoughtful comments and suggestions have improved the quality of this work.

I also thank all faculty in Industrial Engineering Department, for their devoted instruction.

My special gratitude goes to my family and friends. I thank my parents for their unfailing support and unselfish love. Most importantly, I would like to extend my deepest thanks to my beloved wife, Bingyu Ling, for her love and understanding. Her support and encouragement was in the end what made this dissertation possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Simulation-based optimization has attracted great interest among researchers and engineers in recent years due to its wide applications in system design and operation. Especially at a time when there has been an increasing trend in system complexity and uncertainty, traditional optimization techniques cannot be employed directly to handle noises incurred from system performance measurements; in this case, simulation-based optimization is the best way to obtain sound solutions.

Simulation-based optimization is an emerging research area which integrates two traditional tracks of operations research: optimization and discrete-event simulation. It is used to find the optimal system configurations that maximize the system performance or minimize the operating cost, where closed-form expressions of the performance functions or cost functions are not available. Typically, this problem is formulated as follows:

$$\min_{\theta \in \Theta} f(\theta), \tag{1.1}$$

where θ is the design parameter (or configuration) of the system, Θ is the feasible set for θ, and the objective function $f(\theta)$ is usually the expected value of some system performance measure. In the framework of simulation-based optimization, the system is often viewed as a "Black box." Even though we don't have too much information

about the system structure, we can run the simulation and obtain the performance measurements we are interested in, given any input configurations. For instance, the objective function $f(\theta)$ can be estimated by using n independent simulation runs under the same value of θ,

$$\bar{f}(\theta) = \frac{1}{n}\sum_{i=1}^{n} F_i(\theta),$$

where $F_i(\theta)$ is a simulation observation of $f(\theta)$. The simulation output is then used as the input of an optimization procedure to generate better solutions. The relationship between simulation and optimization is illustrated in Figure 1.1.

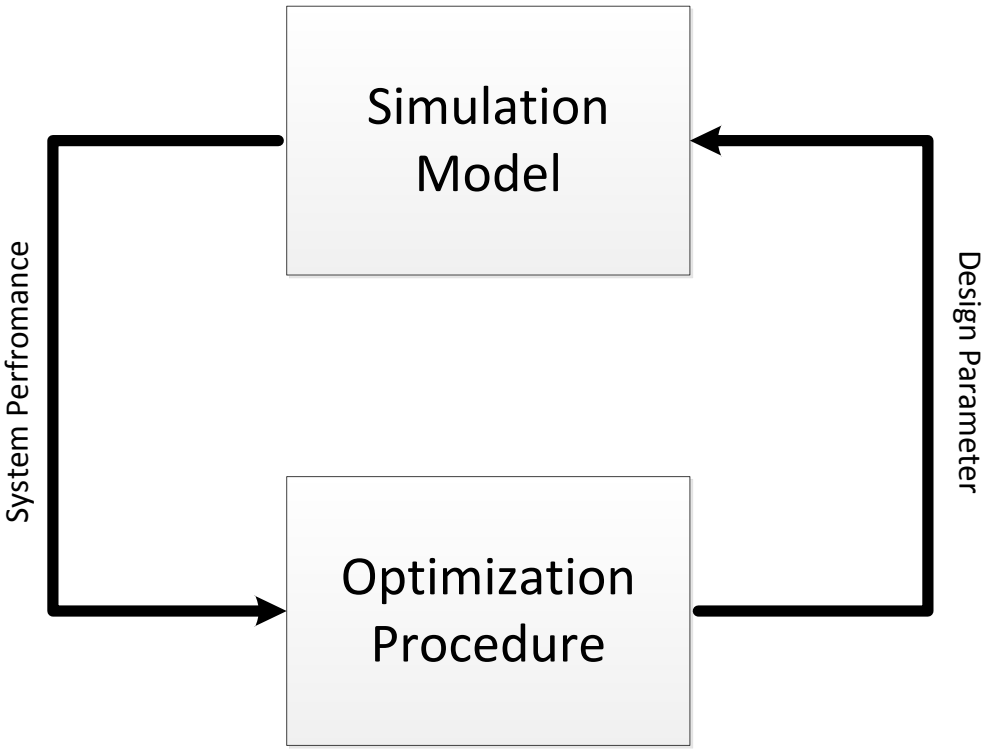

Figure 1.1: The relationship between simulation and optimization.

1.1. Motivation

In many practical situations, the design parameters of the system can only take finite or countably infinite discrete values in the feasible region. For example, when a hospital designs the staffing level in an emergency room to minimize the operation cost, the number of nurses scheduled should acquire a non-negative integer value. This type of problem is usually difficult to solve since the cardinality of the search space will increase exponentially as the dimension of the input parameter is increasing. In addition, it is natural to incorporate constraints into simulation-based optimization problems since most real world systems are constrained by resources. Nevertheless, if the constraints are also very difficult to measure, i.e., they have to be evaluated through simulation, the problem becomes more challenging. The operational difficulty results from uncertainty about solution feasibility, while such an uncertainty is caused by noisy observations of constraint functions via simulation. Consider, for example, an inventory system with discrete demand processes. In this instance, we are required to find the optimal ordering policy by choosing some control parameters, where the goal is to minimize the average ordering and holding costs per unit time while achieving a prescribed level of customer service simultaneously. In such a context, the demand might be random at different time periods and the lead time to receive a order might also be random. So we cannot build a closed-form mathematical model to describe the system behavior. As a result, evaluation of the average cost per unit time and the service level requires the use of a simulation of the system. The optimization of this simulated system is very changeling because the inventory control parameters may be restricted to be discrete and a wide range of potential candidates can be selected. Moreover, the noisy simulation observations of the service level make it very hard to decide whether an input control parameter is feasible or not, since we employ stochastic simulation to estimate the true service level and the sample size is quite

limited. Therefore, we need a more effective method to accommodate the discrete decision variables and noisy constraints in simulation-based optimization.

For some simulation-based optimization problems, the simulation response functions are often known to have certain shape restriction such as convexity. If this is the case, a class of simulation response surface methods can be used to construct a convex function to fit the simulation response and attain the optimal solution by solving a resulting convex optimization problem. The key part in this procedure is to build the simulation response surface, i.e., to fit the convex function effectively, so that the optimization algorithms based on the fitted function can work well to search for optimal solutions. One popular way to fit the convex function based on noisy simulation observations is to find a function which minimizes the sum of squared regression errors and satisfies the convexity constraints by solving a quadratic program. While such a convex regression estimator usually enjoys favorable statistical properties, the computational burden is quite high. When the given number of data points in the simulation output dataset is very large, the quadratic program becomes intractable as the number of constraints becomes even larger. As a result, current available methods can only handle a convex regression problem with a few hundred input data points. Recent studies show, however, that there is a growing need for fitting a convex function to large-scale data. For example, the power, gain, or bandwidth of integrated circuits is often approximated as a convex, or concave, function in the sizes of the transistors contained in integrated circuits (del Mar Hershenson et al., 2001). In this context, more than a few thousand data points can be used to estimate the performance measure of the integrated circuits as a convex function. Hence, a computationally efficient way to fit the convex regression is necessary to address complex problems under real circumstances.

In the simulation-based optimization framework, the measurement of the simulation output has a great effect on the optimization algorithms. In order to make the

algorithms converge quickly and provide high quality solutions, we often need to get accurate estimates of the simulation output. Frequently, when applying simulation-based optimization, we are interested in estimating the steady-state performance of a system. Due to the lack of information on steady-state distribution, however, practitioners usually start the simulation in some arbitrary fashion. This generally leads to an initial transient phase prior to steady-state. Consequently, the inclusion of the initial transient data often leads to bias when estimating stead-state parameters.

The most commonly used method to resolve the initial transient problem is to allow the system to run for a warm-up period and output data during this period are not collected. There is, however, a trade-off when deciding how long the warm-up period should be. If the warm-up period is too short, initialization bias will be introduced to the parameters estimation. If it is too long, the collected data are wasteful and more observations are needed to ensure estimating precision. Various methods have been proposed to detect the length of warm-up period in the simulation literature. These methods usually fall into five categories: graphical methods, heuristic approaches, statistical methods, initialization bias tests and hybrid methods (Hoad et al., 2008).

1.2. Contribution

To address the challenging issues of applying simulation-based optimization to solve practical problems, we propose methods to handle optimizing with discrete decision variables and noisy constraints, to compute the convex response surface function, and to detect the initial transient period in steady-state simulation. In particular, the main contributions of this dissertation can be summarized as follows.

1.2.1 A dual framework and domain extension

To handle discrete decision variables and noisy constraints arising from simulation-based optimization, we investigate the optimization problem of the form:

$$\begin{aligned} \min_{\theta \in C \cap \mathbb{Z}^d} \quad & f^0(\theta) \\ \text{s/t} \quad & f^i(\theta) \leq 0, \quad 1 \leq i \leq r, \end{aligned} \tag{1.2}$$

where C is a nonempty closed convex subset of $\mathbb{R}^d$, and $f^i : \mathbb{Z}^d \to \mathbb{R}$ $(0 \leq i \leq r)$ has no analytic form and thus must be computed only through simulation at each θ in $\mathbb{Z}^d$. Such functions f^i often arise in the setting of a complex stochastic system where one performance measure is described by f^0 and the other performance measures are denoted by f^i $(1 \leq i \leq r)$.

In order to overcome the obstacle of evaluating the constraints based on noisy simulation observations, we seek to incorporate the constraints into the objective function by using the Lagrangian formulation, i.e., the dual formulation, of the original constrained problem:

$$\max_{\lambda \in \mathbb{R}^r_+} \min_{\theta \in C \cap \mathbb{Z}^d} L(\theta, \lambda), \tag{1.3}$$

where the Lagrangian function $L(\theta, \lambda)$ is defined by

$$L(\theta, \lambda) = f^0(\theta) + \sum_{i=1}^{r} \lambda^i f^i(\theta)$$

for $\theta \in \mathbb{Z}^d$ and $\lambda = (\lambda^1, \ldots, \lambda^r) \in \mathbb{R}^r_+$. We observe that the minimizer of the constrained problem (1.2) can be found by finding the optimal solution of (1.3), i.e., the saddle point of $L(\theta, \lambda)$. Since (1.3) is an unconstrained $max - min$ problem, we can use a gradient-based method to solve it. In particular, we wish to apply stochastic approximation to L (and hence update θ in the steepest descent direction

and λ in the steepest ascent direction in each iteration) in search of the saddle point of L. We notice that stochastic approximation requires gradient estimates of L in each iteration, but θ is integer-valued; therefore, the gradient of L with respect to θ cannot be defined in a traditional way. To overcome this obstacle, we extend f^i from a discrete domain to a continuous domain, use the extended f^is to construct the extended L, and then compute the gradient of the extended L in the usual way. The gradient of the extended L is then used in each iteration of stochastic approximation in search of the solution to (1.3). We prove that this procedure is convergent to the optimal solution of the original problem (1.2) almost surely (a.s.) under suitable conditions.

1.2.2 A least absolute deviations based convex estimator

To overcome the computational inefficiency of the least squared errors-based convex regression estimator, we propose to use a formulation which minimizes the sum of absolute deviations instead of the sum of squared errors. This formulation is beneficial from a computational point of view since we can reduce the problem to a linear program, which is easier to handle than a quadratic program. Another advantage of using this formulation is that the resulting convex estimator can provide more robust results when many outliers are present in the input dataset. In this dissertation, we investigate how to build a linear program based on the least absolute deviations criterion. We also provide an efficient algorithm based on the Dantzig-Wolfe decomposition principle to compute the convex estimator. We then establish the statistical consistency of the least absolute deviations estimator by giving a complete proof.

1.2.3 A statistical test based procedure

In this dissertation, we consider the problem of computing the steady-state mean of a system performance in the situation where it is not clear how to initialize the

simulation. In this case, the simulation may start at some arbitrary position which is usually atypical of steady-state behavior. This particular initialization will induce an initial transient phase that introduces a severe bias in the point and interval estimates. To overcome this problem, we will allow the system to warm up before output data are collected and thus eliminate the observations until steady-state behavior becomes apparent. We then compute the arithmetic mean of the remaining observations as an estimate of the steady-state mean.

In our proposed method, the truncation point after which the observations are retained for steady-state analysis will be chosen according to the following techniques:

1. Divide the simulation output into small batches.

2. Obtain the empirical distribution function within each batch.

3. Compare these empirical distribution functions for a change in distribution functions.

1.3. Organization

The dissertation is organized as follows.

Chapter 2 gives a literature review on simulation-based optimization methods with continuous decision variables, discrete decision variables, and noisy constraints. We also provide reviews on some popular methods to fit convex functions and detect the initial transient period in steady-state simulation.

Chapter 3 presents the details of our proposed dual formulation to deal with the stochastic constraints in simulation-based optimization. We also discuss some possible ways to extend the optimization program from a discrete domain to a continuous domain and give a specific algorithm to iteratively search the optimal solution based on stochastic approximation. We prove the convergence of our proposed algorithm

and demonstrate its applications in practical problems. Examples will be given, including, but not limited to: optimizing the ordering policy in a stochastic inventory system; staffing a call center, and staffing an emergency room with random service times and a complex arrival process.

Chapter 4 describes an alternative formulation of the convex regression problem. We show how the proposed least absolute deviations-based estimator can be computed from an equivalent linear program. Furthermore, we study the unique property of this linear program by considering its dual problem, which exhibits a block-angular form in its constraints. Then we discuss the solution of this linear program based on a Dantzig-Wolfe decomposition procedure. We present numerical examples to illustrate the relative performance of the proposed estimator compared to that of the least squares estimator. We also establish the consistency of the proposed estimator and its derivative by proving that, under modest assumptions, the estimator and its derivative converge almost surely (a.s.) to the true values as the number of data points increases to infinity.

Chapter 5 introduces a framework of using the Kolmogorov-Smirnov test method to detect the warm-up period in steady-state simulation and provide better estimators for simulation performance measures. We display the efficiency of the proposed method by comparing it with some widely used initialization procedures. The comparisons are conducted using examples from queueing systems and inventory systems.

Chapter 6 summarizes the contribution of this dissertation and outlines some interesting future research directions.

Chapter 2

Literature Review

In this chapter, we give a comprehensive review on popular simulation-based optimization techniques. A brief discussion on the convex regression problem and its potential application in simulation-based optimization is also provided. Since the initial transient problem is common in most stead-state simulations, we also review some widely used methods to detect the initial transient phase.

2.1. Simulation-based Optimization

Simulation-based optimization is one of the fastest growing research areas in the operations research society during the past two decades. Numerous studies have been conducted to obtain optimal or good enough solutions for simulation-based optimization problems efficiently with limited computation efforts.

Simulation-based optimization methods can be categorized based on the nature of the problem structure. If the feasible region of the optimization problem is a continuous set, then it may be possible to use a metamodel-based optimization method or a gradient-based method such as stochastic approximation (Robbins and Monro, 1951) to search the solution iteratively. If the decision variable is discrete and the solution space is large, then random search method (Andradóttir, 1995) and metaheuristics

(Ólafsson, 2006) may be appropriate to solve the desired problem. If the feasible region has a small number of candidate solutions, say less than 100, then some statistical analysis based methods such as ranking and selection (Goldsman and Nelson, 1998) can be applied to find the best solution with the smallest mean objective value. Figure 2.1 shows the classification of simulation-based optimization methods based on problem structure. For more comprehensive reviews on general simulation-based optimization methods, see Henderson and Nelson (2006), Ólafsson and Kim (2002), Fu (2002), Nelson (2010), Carson and Maria (1997), Hong and Nelson (2009), and Swisher et al. (2004).

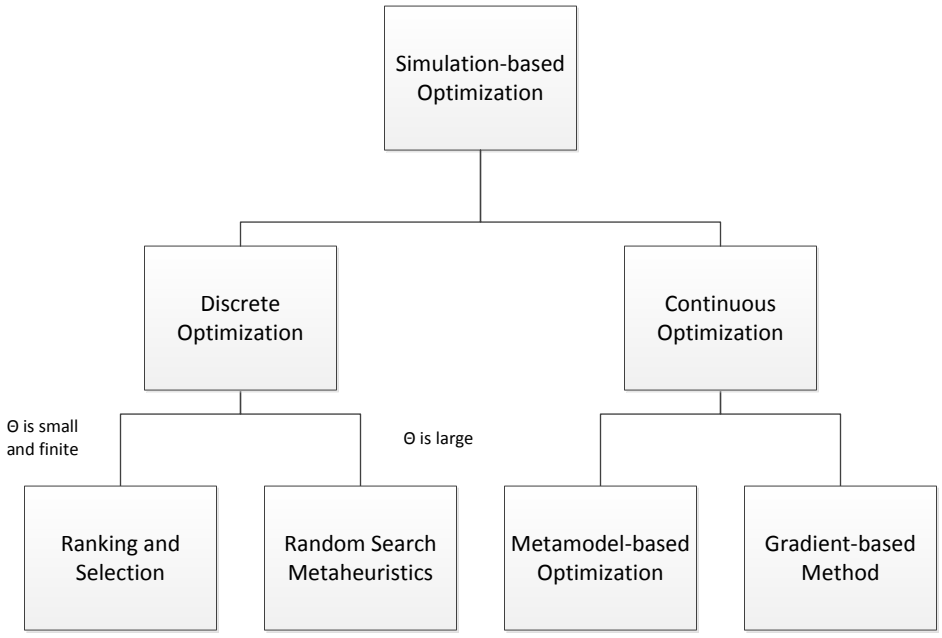

Figure 2.1: Simulation-based optimization method classification

In this chapter, we briefly review some popular simulation-based optimization methods according to the above classification criteria. Section 2.1.1 and Section

2.1.2 present methods designed for solving both continuous and discrete simulation-based optimization problems, respectively. In Section 2.1.3, we focus on methods to solve simulation-based optimization problems with noisy constraints. The purpose of this chapter is to show that although many efficient methods are available for general simulation optimization problems, few of them address the situation when noisy constraints present. The presence of noisy constraints do increase the difficulty of simulation-based optimization problems. Specialized algorithms and theories are needed to fill the gap between unconstrained simulation-based optimization methods and constrained simulation-based optimization methods.

2.1.1 Continuous Simulation-based Optimization Methods

Simulation-based optimization with continuous decision variables is one of the most frequently studied area in the literature. Some methods fall into the category of gradient-based methods, which try to obtain gradient information through simulation directly, while others use an indirect strategy, which constructs a metamodel based on simulation observations and computes gradient estimates through the metamodel. Since our main focus is the discrete simulation-based optimization problem, we briefly review some popular methods in both categories. Nonetheless, we want to point out that our proposed method is based on the framework of gradient-based methods, even though our method is designed to treat discrete decision variables and noisy constraints. Accordingly, we also present the general procedures and formulations of some gradient-based methods for later reference.

2.1.1.1 Gradient-based Method

Gradient-based optimization methods are widely adopted in deterministic optimization to iteratively search for a minimum of the objective function. When applying this method to simulation-based optimization problems, one faces the difficulty of

estimating the stochastic gradient since the simulation output is random. Fu (2006) surveys the main approaches available in the literature, including finite differences, simultaneous perturbations, perturbation analysis, the likelihood ratio/score function method, and weak derivatives.

Stochastic approximation (SA) is the stochastic version of the steepest descent method in nonlinear optimization. It iteratively searches from one solution to another in the direction of the estimated gradient, since the closed form of the gradient doesn't exist. First invented by Robbins and Monro (1951) and Kiefer and Wolfowitz (1952) as a root-finding procedure, the method has received extensive attention in the past five decades. For a detailed study of SA method and its application, see Kushner and Yin (2003) and Fu (2006).

In general, the SA method uses the following recursion to update solution iteratively:

$$\theta_{n+1} = \Pi_\Theta \left(\theta_n - a_n \widehat{\nabla} f(\theta_n) \right),$$

where θ_n is the solution at iteration n, a_n is a sequence of positive real numbers, $\widehat{\nabla} f(\theta_n)$ is a gradient estimate of the objective function f with respect to θ at θ_n, and Π_Θ is an operator which projects a solution outside of the feasible region Θ back into Θ. It is well known that SA type methods have nice convergent properties under certain assumptions (e.g., $\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$). But in practice, the performance of SA method relies on the choice of a_n. If a_n is too small, the algorithm converges very slowly, i.e., θ_n obtained at iteration n does not change too much comparing with θ_{n-1}. On the other hand, if a_n is too large, the algorithm becomes quite unstable since oscillations occur (Fu, 2006). Another important issue associated with the performance of SA method is the quality of gradient estimates. There are several ways to estimate the gradient without any prior knowledge about the simulated system structure. Two of the most famous ones are the finite differences

(FD) method and the simultaneous perturbation stochastic approximation (SPSA) method.

2.1.1.1.1 Finite Differences The FD method was first introduced by Kiefer and Wolfowitz (1952) to determinate the gradient in the SA algorithm. It uses a straightforward way to estimate the gradient by taking small perturbations at each dimension of decision variable θ. The perturbation can be taken for either one side or both sides. A one-sided forward difference gradient estimator of $\nabla f(\theta)$ can be denoted by

$$\widehat{\nabla} f(\theta) = \frac{1}{h} \begin{pmatrix} F(\theta + he_1) - F(\theta) \\ F(\theta + he_2) - F(\theta) \\ \vdots \\ F(\theta + he_d) - F(\theta) \end{pmatrix},$$

where h is a small positive value, e_i is a unit vector with a 1 in the i^{th} position and 0's elsewhere. For the forward FD formulation, $d + 1$ simulation runs are needed to obtain one such estimator. A two-sided FD estimator can be denoted by

$$\widehat{\nabla} f(\theta) = \frac{1}{2h} \begin{pmatrix} F(\theta + he_1) - F(\theta - he_1) \\ F(\theta + he_2) - F(\theta - he_2) \\ \vdots \\ F(\theta + he_d) - F(\theta - he_d) \end{pmatrix}$$

For the two-sided FD estimator, $2d$ simulation runs are required to compute one such estimate.

When optimizing a complex stochastic system, the dimension of θ, i.e, d, is often very large. Obviously, the FD method is not an efficient way to estimate the gradient, since the number of simulation runs required to obtain one gradient estimate is proportional to the dimension of θ.

2.1.1.1.2 Simultaneous Perturbation Stochastic Approximation To overcome the inefficiency of the FD estimator for large dimensional problems, Spall (1992) designs a simultaneous perturbation stochastic approximation method which requires only 2 simulation runs to compute a gradient estimate no matter how large d is. In the SPSA method, the ith component of the gradient estimator can be obtained by

$$\widehat{\nabla}_i f(\theta) = \frac{F(\theta + h\Delta) - F(\theta - h\Delta)}{2h\Delta_i}$$

where $\Delta = (\Delta_1, \ldots, \Delta_d)$ is a d-dimensional vector, Δ_is are independent and identically distributed (i.i.d.) random variables taking values $+1$ or -1 with equal probability. Thus, the SPSA estimator only requires two simulation runs, i.e., at $\theta + h\Delta$ and $\theta - h\Delta$, to compute the gradient estimate. Since the evaluation of performance functions is quite expensive in most simulation-based optimization problems, SPSA has been considered an efficient way to estimate the gradient when direct information about the gradient is unavailable. It has been shown that the SPSA algorithm and the FD stochastic approximation algorithm can achieve the same level of statistical accuracy under some general conditions and a given number of iterations, while the number of function evaluations in SPSA is only $1/d$ times of that in the FD stochastic approximation algorithm (Spall, 2003). For a detailed study of SPSA method for simulation-based optimization, see Spall (1998, 1999), Fu and Hill (1997).

In addition to indirect gradient estimation method such as FD and SPSA, there are some direct methods in the literature which take advantages of the structural knowledge of the simulation model to acquire more gradient information. Such examples include Perturbation Analysis (PA) (Ho and Cao, 1983; Fu and Hu, 1997) and Likehood Ratio (Reiman, 1989; Glynn, 1990). For a comprehensive review of the research for these methods, see Fu (2008).

2.1.1.2 Metamodel-based Method

Metamodel-based method is a general simulation-based optimization method which attempts to find an approximation to fit the simulation input-output response function. Since the approximation is constructed based on simulation experiments, it is often referred to as *metamodel* (Kleijnen, 1975, 2008b). Once the metamodel is constructed based on some simulation observations, the simulation-based optimization problem is actually simplified to a deterministic optimization problem, which subsequently can be solved by many efficient deterministic optimization algorithms. For a detailed introduction to this method, see Barton and Meckesheimer (2006) and Kleijnen (2008b).

One major task of metamodel-based method is to create the so-called metamodel. One of the most well-known metamodels is the response surface metamodel. It was first brought forth by Box and Wilson (1951) for the purpose of designing the optimal operating conditions for some chemical processes. This type of metamodels uses first or second-order polynomial functions to fit the simulation observations. Therefore, it is more appropriate for local approximation in most situations (Barton, 1992; Barton and Meckesheimer, 2006). More recently, many researchers have successfully applied spatial correlation (kriging) metamodels for both deterministic simulations (Simpson et al., 1998; Booker et al., 1999) and stochastic simulations (Mitchell and Morris, 1992; Ankenman et al., 2010) to overcome the limitations of response surface metamodels and provide more flexible modeling techniques. Barton and Meckesheimer (2006) summarize some commonly used metamodel functions for simulation optimization along with comments on experiment designs and global and local properties (Barton, 2009).

Typically, after choosing the metamodel form, one needs to design simulation experiments to obtain observations, fit the metamodel using data, and conduct optimization using the metamodel (Barton and Meckesheimer, 2006). Two strategies

have been often suggested to conduct metamodel-based optimization: iterated local metamodels and global metamodel fits. The former strategy uses linear or quadratic regression models to find the response function for local data. The response function is then employed to estimate the search direction toward optimum response and establish a new local area. The global metamodel fit strategy usually uses spline, kriging metamodel, natural network or radial basis function to accommodate the global data. Deterministic global optimization method is then applied to detect the global optima based on the metamodel. Instead of conducting the optimization procedure iteratively, the global strategy only runs the optimization one time.

2.1.2 Discrete Simulation-based Optimization Methods

The problem of minimizing an unconstrained function over a discrete set has gained a considerable amount of attention from the research community, and a number of methods are proposed in the literature; see Goldsman and Nelson (1994) Andradóttir (1995), Yan and Mukai (1992), Ho and Vakili (1992), Shi and Ólafsson (2000), Hong and Nelson (2006), Kleywegt et al. (2001), Gelfand and Mitter (1989), Glover (1989), Liepins and Hilliard (1989), Gerencsér et al. (1999), Gokbayrak and Cassandras (2002), Dupač and Herkenrath (1983), and Prudius and Andradóttir (2009) for example. For more comprehensive surveys, see Nelson (2010), Henderson and Nelson (2006), Swisher et al. (2004), and Fu (2002).

2.1.2.1 Ranking-and-Selection

Ranking-and-selection (R&S) methods are designed to solve simulation-based optimization problems when the number of alternative systems is finite and small. The main focus of this method is to select a system with the best expected value of performance based on a pre-specified probability of correct selection. See Kim and Nelson (2006), Hong and Nelson (2009), and Nelson (2010) for a detailed introduction.

In this method, $k \geq 2$ feasible solutions, $\theta_1, \theta_2, \ldots, \theta_k$ are considered. The jth simulation observation at θ_i, $F_j(\theta_i)$, is assumed to be normally distributed with mean $f(\theta_i)$ and standard deviation σ_i^2. Denoting the index of the best and the second best system by k and $k-1$ respectively, R&S methods are valid if

$$Pr\{\text{select solution } \theta_k | f(\theta_k) \leq f(\theta_{k-1}) - \delta\} \geq 1 - \alpha,$$

where $\delta \geq 0$ is the indifference-zone(IZ) parameter and $1 - \alpha$ is the pre-specified probability of correct selection. δ is used to decide the simulation runs required to achieve the $1 - \alpha$ significance level. The earliest and simplest R&S method is Bechhofer's procedure (Bechhofer 1954), which uses a sample size

$$n = \lceil \frac{2h^2\sigma^2}{\delta^2} \rceil,$$

for each θ_i. h in the above formula denotes the $1 - \alpha$ quantile of the maximum of a multivariate normal random vector $(Z_1, Z_2, \ldots, Z_{k-1})$ with means 0 and variances 1. σ^2 is assumed to be known and $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2 = \sigma^2$. The procedure then computes the sample mean of $F_j(\theta_i)$ and selects the one with the best mean performance. When the variance σ^2 is unknown to us, some two-stage procedures are often used, where the first stage estimates the variances based on some preliminary samples. Such procedures include Rinott's procedure (Rinott, 1978), Nelson and Matejcik's procedure (Nelson and Matejcik, 1995) and NSGS procedure (Nelson et al., 2001). What's more, when the simulation data samples are collected sequentially, some full-sequential procedures are also proposed (Paulson, 1964; Kim and Nelson., 2001; Kim and Nelson, 2006). Such procedures usually maintain a solution pool and take one observation for each solution that is in the pool at each iteration. Then the mean performance is evaluated at each solution based on all the observations obtained so far. The procedures then drop some solutions from the pool by comparing their

mean performances with a calculated deviation bound. The iteration continues until there is only one solution left in the pool.

Although R&S method is originally designed to find the best system among a small and finite alternatives, it has been embedded in other simulation-based optimization algorithms in order to improve their performances. For example, many discrete simulation-based optimization algorithms require sampling good candidate solutions from the neighbourhood of the current solution, where the neighbourhood is often small and R&S might be used to select the best candidate (Pichitlamken et al., 2006). Another application of R&S is to select the best solution among all the solutions visited by an iterative simulation-based optimization algorithm (Boesel et al., 2003).

2.1.2.2 Random Search

Random search method is one of the most widely used methods to solve discrete simulation-based optimization problems when the feasible region Θ is large or countable infinite. See Andradï¿½ttir (1998), Swisher et al. (2004), Andradï¿½ttir (2006) and Hong and Nelson (2009) for a comprehensive survey.

The basic idea of this method is to sample a few candidates from the neighbourhood of the current best solution and simulate the performance. The solution with the best mean performance then is chosen for the next iteration. The general procedure of such method is shown as follows:

Generic random search algorithm for simulation optimization (Andradï¿½ttir, 2006):

Step 0. Initialize: Choose the initial sampling strategy S_1. Set $n = 0$.

Step 1. Sample: Select $\theta_n^{(1)}, \ldots, \theta_n^{(M_n)} \in \Theta$ according to the sampling strategy S_n.

Step 2. Simulate: Estimate $f(\theta_n^i)$, for $i = 1, \ldots, M_n$, using simulation.

Step 3. Update: Use the simulation results obtained so far in Step 2 to compute an estimate of the optimal solution θ_n^* and to choose an updated sampling strategy S_{n+1}. Let $n = n + 1$ and go to Step 1.

There are multiple versions of random search methods in the literature. These available algorithms can be categorized based on the neighbourhood structure, sampling distribution, evaluation scheme and the method of estimating the optimal solution (Hong and Nelson, 2009).

Yan and Mukai (1992) propose the stochastic ruler method which samples a potential solution by comparing its mean performance with a uniform random variable (stochastic ruler). The method is proved to converge in probability but the finite-time performance is not desirable due to increasing simulation efforts. Alrefaei and Andradï¿½ttir (2001) introduce a modified stochastic ruler method that uses the number of times the method visits every state to estimate the optimal solution. Instead of using an increasing number of estimates of the objective function values per iteration, this method sets a fixed number of such estimates at each iteration. And the authors also show that the algorithm converges almost surely to the set of global optimal solutions under more general conditions.

Andradóttir (1995) utilizes a random walk approach to develop a simulation-based optimization algorithm over a large discrete feasible region. In each iteration of this algorithm, the values of the objective function at the current point and neighbouring feasible points are estimated via simulation, and the alternative that yields the better

estimate is used for the next iteration. The author also shows the local convergence of this algorithm when adopting the feasible alternative that has been visited most often in the iteration process to estimate the optimal solution. And the globally convergent version of this algorithm is presented in Andradï¿½ttir (1996).

Alrefaei and Andradï¿½ttir (1999) present a simulated annealing method with constant temperature and use the the most frequently visited solution by the algorithm as an estimate for the optimal solution to increase the convergence speed. Gong et al. (1999) propose the stochastic comparison algorithm to handle discrete simulation-based optimization problems with unstructured solution space. Unlike the simulated annealing method, this method does not require well designed neighbourhood structures to guarantee the convergence. Andradï¿½ttir (1999) submits a variant of the stochastic comparison method and uses the solution with the best estimated objective value as the estimate for the optimal solution.

Shi and Ólafsson (2000) introduce the idea of searching in the most promising region and propose the nested partitions (NP) method. At each iteration of this method, the current most promising region is partitioned into several subregions. The method then randomly takes samples from each subregion and the entire surrounding region and evaluates the performances based on simulation. After calculating the promising index for each region, the method either chooses one subregion as the new most promising region or moves backward to a larger region that contains the old most promising region. This method is proved to converge almost surely.

Hong and Nelson (2006) propose a version of random search methods called Convergent Optimization via Most-Promising-Area Stochastic Search (COMPASS), which utilizes a unique neighbourhood structure and results in a provably convergent algorithm to a local optimal solution. In this method, the most promising area is defined as the region where feasible solutions are closer to the current best solution

than to other solutions that the algorithm has visited so far based on Euclidean distance measure. Such an area is fully adaptive. Hong et al. (2010) suggest using a coordinate sampling strategy to speed up COMPASS algorithm for high-dimensional problems, where a new sampled solution differs from the current solution in only one dimension.

2.1.2.3 Metaheuristics

Metaheuristic is an iterative method aimed at improving the search based on some solution quality measures. Methods used in this category usually do not presume any structure knowledge on the problem to optimize and can solve those problems with large feasible region. However, there is no guarantee that this type of method can converge to the optimal solution. But they might find a sound solution very quickly and perform quite well in practice. Although metaheuristics are designed to solve deterministic combinatorial optimization problems, they have been successfully applied to solve simulation-based optimization problems. Some metaheuristic methods include genetic algorithm (GA), tabu search (TS), scatter search (SS), and particle swarm optimization (PSO). For a comprehensive review on metaheuristics for simulation-based optimization, see Ólafsson (2006) and April et al. (2003).

Genetic algorithm was invented by Holland (1975) as a global optimization approach. GA uses the general framework of random search method and adopts an innovative way to construct the neighbourhood. The algorithm begins with a population of solutions instead of a single solution. At each iteration, a number of solutions in the population are selected based on a fitness function defined by some pre-specified strategies (e.g., top n strategy and roulette strategy). These selected solutions are then used to generate new candidates based on crossover and mutation operators. The crossover operator chooses two solutions with high fitness values as parents, randomly selects a segment from each parent and then exchanges the segment. The

mutation operator chooses only one solution and randomly modifies one sub-segment of this solution. Examples of using GA in the simulation-based optimization setting include Tompkins and Azadivar (1995), Ishibuchi and Murata (1996), Vavak and Fogarty (1996), Paul and Chanev (1998), etc. Note that GA is widely used in many commercial simulation-based optimization software packages, including AutoStat for AutoMode and SimuRunner for ProModel (Fu, 2002).

Tabu search is another popular metaheuristic used for simulation-based optimization (Glover and Laguna, 1997). It differs from the traditional random search method by maintaining a tabulist. This tabulist stores the solutions that have been visited in the recent k iterations and is updated iteration by iteration. When one decides the best candidate from the neighbourhood of current solution, all solutions in the neighbourhood are evaluated and the one with the best simulated performance is selected. The algorithm then moves to this best candidate as long as it is not in the tabulist, even though this candidate solution might be inferior to the current solution. This feature enables the search to escape local optima and prevents cycling. It is notable that the tabu search method has been integrated to some commercial simulation-based optimization software such as OptQuest (April et al., 2003) and Optimizer for WITNESS (Fu, 2002).

Scatter search (Glover, 1977) is another population based metaheuristic and is often used with the tabu search method together. The basic idea is to select a population of solutions from previous solution efforts and use them as reference points. New feasible solutions are then generated by using the linear combinations of the reference points and mapping them into the feasible region. This method is also embedded in some modern commercial software, e.g., OptQuest (Fu, 2002).

Like GA and SS, particle swarm optimization (PSO) is another population based simulation-based optimization method. It was originally proposed by Kennedy and Eberhart (1995) and inspired by social behavior of animals, e.g., bird flocking and

fish schooling. It also begins with a population of randomly selected solutions. Each potential solution, which is called a particle, computes its coordinate relative to the best solution (fitness) obtained so far. A particle can also share information of its best position with other particles in the neighbourhood and then change its own velocity towards the best local position. As a result, when a particle takes all other solutions in the population as topological neighbouring solutions, the best local solution is also the global best solution. For a detailed study of PSO in the simulation-based optimization context, see Kennedy et al. (2001).

2.1.3 Simulation-based Optimization Methods with Constraints

Although many methods have been proposed to solve the general simulation-based optimization problems, a limited number of methods are available in the literature to deal with noisy constraints.

In the presence of one stochastic constraint, Andradóttir et al. (2005) and Andradóttir and Kim (2010) propose a two-phase R&S procedure where the first phase identifies all feasible solutions or near-feasible solutions with a pre-specified probability of correct identification, and the second phase solves the problem of interest with the solutions identified in the first phase. Batur and Kim (2001) then extend the R&S procedure to the case of multiple constraints while Pujowidianto et al. (2009) address how to allocate computer time in an optimal way among the solutions in order to maximize the probability of correctly identifying the optimal solution. These methods have a requirement that all the solutions must be simulated as least once, so they are more appropriate to the setting where the domain of f^0 is finite and contains a small number of elements.

In the presence of multiple constraints, the idea of replacing a constrained optimization problem with an unconstrained one by adding a penalty function to the

objective function has been investigated; Li et al. (2009) combine a penalty function type method with a random search scheme, and Whitney et al. (2001) incorporate a penalty function type method into a gradient-based search scheme. Hill et al. (2003) propose a version of the Simultaneous Perturbation Stochastic Approximation (SPSA) method that can be applied to cost functions defined on discrete sets. However, the convergence of these methods is not guaranteed or is based on restrictive assumptions that are difficult to verify.

Kleijnen (2008a) summarizes the generalized response surface methodology, which selects one simulation response as goal and the others as constrained variables. Unlike the steepest ascent method used by traditional RSM, this method combines the gradients that are based on local first-order polynomial approximations with Mathematical Programming to estimate a better search direction. A bootstrap procedure is then used for testing whether the estimated solution is indeed optimal or not.

Ahmed et al. (1997) and Alkhamis and Ahmed (2005) use the concept of hypothesis test to handle the stochastic constraints and combine the hypothesis-test-based criterion with some random search schemes. When deciding whether a solution in the neighbourhood is feasible or not, they test H_0: $f_i(\theta) \leq 0$ against the alternative hypothesis H_1: $f_i(\theta) > 0$. They consider θ is feasible if the lower bound of the confidence interval computed for $f_i(\theta)$ at certain pre-specified significant level is less than or equal to 0.

Another way of transforming a constrained problem to an unconstrained one is to use the Lagrangian function. The idea of using the Lagrangian function has already been adopted when the decision variables are continuous (see Kushner and Sanvicente (1975) and p. 177 of Kushner and Clark (1978) for example), but this idea has never been explored in the setting of discrete decision variables. This dissertation explores this idea in the discrete setting and studies the effectiveness of this approach. One of our motivations is that in the deterministic optimization problem, the Lagrangian

method has certain advantages over the penalty function type methods because most penalty function type methods suffer from numerical instabilities as the controlling parameter becomes too large or too small, see Murray (1967) for example. We present this methodology in details in Chapter 3 and propose the framework of applying Lagrangian function method to solve discrete simulation-based optimization problems.

2.2. Convex Regression

In the second part of this dissertation, we aim to study the problem of estimating a multivariate regression function under a certain shape restriction such as convexity. This problem is usually referred to as convex regression in the literature.

2.2.1 Applications of Convex Regression

Convex regression has wide applications in both economics and operations research. In economics, a production function or a customer utility function is often estimated by fitting a convex function to the empirical data; see, for example, Skiba (1978) and Meyer and Pratt (1968). In the operations research setting, various performance measures of stochastic models in queueing systems and inventory systems have the shape characteristic of convexity. For example, in a single server queue, the mean waiting time of a customer is convex with respect to the mean service times and the inter-arrival times when the service and inter-arrival times are subject to certain probability distributions (Shanthikumar and Yao, 1991); in a queueing system of three single-server stations connected in tandem, the mean sojourn time of a customer is convex with respect to the mean service times at the server stations when the service and inter-arrival times are subject to certain probability distributions (Shanthikumar and Yao, 1991); in a single-item continuous-review (Q, r) inventory system, the

steady-state mean total costs per unit time is convex in the control parameters Q and r when the demand follows certain stochastic process (Zheng, 1992).

Convex regression also has promising applications in the domain of simulation-based optimization. In the case that the simulation response is known to be convex, convex regression can be used to construct the metamodel, i.e., approximate the objective function and constraints based on the noisy simulation observations. Then many deterministic convex optimization methods can be easily applied to find the optimal solution of the metamodel and provide a good approximate solution of the original simulation-based optimization problem. Consider the (Q, r) inventory system mentioned above, our goal is to find the optimal values of Q and r to minimize the mean total costs per unit time. Instead of solving the simulation-based optimization directly, we can simply fit a convex function to approximate the costs function given the observed (Q, r) values and the observed mean total costs. Then we can easily optimize the fitted function based on some gradient-based schemes in convex optimization, since the sub-gradient at each observation is readily available after we run a convex regression procedure. Unfortunately, in many applications of simulation-based optimization, there is no prior guarantee that the true simulation response function is convex. In the absence of convexity, convex regression can fit a convex function that is closest to the true response function in some measure space (Lim and Glynn, 2012). As a result, the convex regression based optimization provides a good heuristic solution or a good start solution for other non-convex optimization procedures.

2.2.2 Univariate Convex Regression

This dissertation is concerned with providing a numerically efficient way of computing the best fit of a convex function and proving the consistency of the proposed method. We are interested in estimating the unknown function $f_* : [0, 1]^d \to \mathbb{R}$ from the

observed data $(X_1, Y_1), \ldots, (X_n, Y_n)$, where

$$Y_i = f_*(X_i) + \varepsilon_i$$

for $i \geq 1$, the X_is are continuous $[0,1]^d$-valued independent and identically distributed (iid) random vectors, and the ε_is are iid random variables with zero median and $\mathbb{E}(|\varepsilon_1|) < \infty$.

When f_* is known to be convex, a natural way to estimate f_* is to minimize the sum of squares

$$\psi_n(g) \triangleq \frac{1}{n}\sum_{i=1}^{n}(Y_i - g(X_i))^2$$

over the set of convex functions

$$\mathcal{C} = \left\{g : [0,1]^d \to \mathbb{R} \text{ such that } g \text{ is convex}\right\}.$$

When ψ_n is used as a goodness-of-fit criterion, the fitted function is referred to as "least squared errors" (LSE) estimator (Hildreth, 1954).

The LSE-based convex regression in one dimension setting ($d = 1$) is well studied both theoretically and computationally. The consistency of the LSE estimator is established by Hanson and Pledger (1976). Many efficient algorithms are also developed to compute such an estimator. These algorithms rely on the special structure of pointwise convexity. Given $x_{i-1} < x_i < x_{i+1}$ for $i = 2, 3, \ldots, n$, the convexity constraints can be enforced by the following conditions:

$$\frac{g(x_i) - g(x_{i-1})}{x_i - x_{i-1}} \leq \frac{g(x_{i+1}) - g(x_i)}{x_{i+1} - x_i}, \quad i = 2, 3, \ldots, n.$$

Then the convex regression problem is converted into a quadratic program with $n-2$

linear constraints (Dent, 1973). See Wu (1982) and Fraser and Massam (1989) for algorithms that solve such an quadratic program with linear convexity constraints. Besides LSE method, some nonparametric methods are also proposed to estimate a convex function. Meyer (2008) suggests a spline-based method and extends it to a convex-restricted regression problem. Birke and Dette (2007) study a kernel regression method which starts with estimating the derivative of the regression function, which is then isotonized and integrated to obtain a strictly convex regression estimator. For some other methods in this category, see Turlach (2005), Shively et al. (2011), and Chang et al. (2007).

2.2.3 Multivariate Convex Regression

When it comes to multiple dimensions setting, less literatures are available. Allon et al. (2007) introduce a nonparametric maximum likelihood method to fit the convex function. Aguilera et al. (2011) propose a two-step smoothing and fitting method which starts from some initial smooth estimator and then obtains the convex estimator by computing the convex hull. Hannah and Dunson (2011) design a nonparametric Bayesian method which estimates the regression function as the maximum of a set of hyperplanes. Kuosmanen (2008) extends the LSE estimator to multiple dimensions case and formulates the minimization problem as a finite quadratic programming (QP) problem. Until recently, the consistency of the LSE estimator for multiple dimensions has been well developed; see Lim and Glynn (2012) and Seijo and Sen (2011). But the LSE estimator suffers from computational inefficiency. Minimization of ψ_n over $\mathcal{C}$ can be formulated as a QP with $(d+1)n$ decision variables and n^2 constraints (Kuosmanen, 2008). The computational burden of solving this QP becomes heavy especially when dn exceeds a few hundred (Lim, 2010). However, recent studies show that the idea of fitting a convex function can be applied to large-scale data. For example, the power, gain, or bandwidth of integrated circuits is often approximated

as a convex or concave function in the sizes of the transistors contained in integrated circuits (del Mar Hershenson et al., 2001). In this context, more than a few thousand data points can be used to estimate the performance measure of the integrated circuits as a convex function. Thus, there is a growing need of fitting a convex function to large-scale data.

2.3. Initial Transient Phase Detection

The initial transient phase detection problem has been widely studied in simulation literature. According to Hoad et al. (2008)'s review, the methods for detecting the initial transient period in steady-state simulation can be classified into five categories: graphical methods, heuristic procedures, statistical methods, initialization bias tests and hybrid methods. Here we present some popular methods in each category and briefly review the advantages and disadvantages of those methods.

2.3.1 Graphic Methods

The methods in this category determine the warm-up period by visually inspecting the time-series data of the simulation output. The most famous one is the Welch's method, which is based on calculating the moving averages of batch means across replications and then plotting them. The general procedure of the Welch's method is described in Law and Kelton (2000). In the procedure, the moving averages are calculated within a windows size w, and w is increased until the plot of the moving average presents smooth trend. Then the warm-up period is viewed as the period before the plot becomes smooth. Although the Welch's method is quite simple and doesn't make any particular assumption on the simulation output, the performance relies on the estimation of several parameters such as run length, number of replications, and the

window size w (Law, 1983). And this method might also overestimating the warm-up method since it is based on calculating cumulative statistics (Pawlikowski, 1990; Wilson and Pritsker, 1978; Roth, 1994). Some alternatives to the Welch's method include CUSUM plots (Nelson, 1992), cumulative-mean plots (Gordon, 1969; Banks et al., 2001), and ensemble average plots (Banks et al., 2001; Pawlikowski, 1990).

Recently, a statistical process control (SPC) based method is also introduced to detect the warm-up period (Robinson, 2002). In such a method, the simulation output data are batched to reduce potential nonnormality and autocorrelation in the time-series. Then a control chart of the batch means is constructed and the warm-up period can be determined by inspecting when the process is in-control and remains in control according to some rules in SPC. The SPC based procedure can be automated and is easy to implement. But it does assume independence and normality of the simulation output data, which might be problematic in some situations even batching technique is used. In addition, the performance is also affected by the way to estimate the mean and variance.

2.3.2 Heuristic Procedures

Heuristic procedures usually provide specific rules for how to identify the warm-up period. Those procedures can be automated and so it is very easy to integrate them with simulation programs.

Conway (1963) proposes an intuitive rule to truncate the data in the warm-up period. In a time-series of simulation output, the first data point that is neither the maximum nor the minimum of the remaining observations is set as the truncation point. And the procedure can be conducted for several replications and the maximum of those truncation candidates is selected. Gafarian et al. (1978) provide a specific procedure to conduct the Conway rule. They also propose a backwards version of the Conway rule. All these procedures are easy to implement in computer programs

and are not dependent on any assumption of the simulation data and estimation of parameters aside from replications. However, such procedures might underestimate or overestimate the warm-up period when they are used in $M/M/1$ system.

Fishman (1973) introduces a rule based on counting the number of times the simulation output data crossing the cumulative mean backwards to the beginning. If the number of crossing reaches a pre-specified threshold, the time-series have reached the truncation point. Of course, the larger the value of this number is, the more confidence we have that the warm-up period has been detected. Gafarian et al. (1978) provide a detailed algorithm that implements this rule. Although the crossing-of-the-mean rule is simple to implement, the performance relies on the appropriate value of the pre-specified threshold, which is very hard to choose.

A marginal standard error rule (MSER) is suggested by White Jnr (1997) to detect the warm-up period. The idea behind this rule is that a data point is viewed from the warm-up period if its impact on calculation of the confidence interval is significant. Given the simulation output time-series $Y_1, Y_2, \ldots, Y_n$, the truncation point is selected at the point d by solving the following unconstrained minimization problem:

$$d^* = \arg \min_{0 \le d \le n} \left[\frac{1}{(n-d)^2} \sum_{i=d+1}^{n} \left(Y_i - \overline{Y}_{n-d} \right) \right],$$

where $\overline{Y}_{n-d} = \dfrac{1}{n-d} \sum_{i=d+1}^{n} Y_i$. However, White Jnr et al. (2000) report that the performance of MSER decreases when the bias increases. They modified MSER by using batches of length five to resolve this issue and showed that the MSER-5 rule performs better than the original one.

2.3.3 Statistical Methods

Kelton and Law (1983) propose to use regression analysis to detect the initial transient period. In their approach, the simulation output data are grouped into several batches

and the batch means are calculated to form a new time-series. They then fit the regression line by using generalized least square regression procedure from the end of the series and move backwards. The procedure proceeds until the slope of the fitted line is significantly different from zero, then all data before the stopping point are identified as the initial transient period data. However, the procedure requires the estimation of nine parameters, including number of replications, initial length of each replication, number of batches, maximum initial deletion proportion, minimum initial deletion proportion, etc. Although Kelton and Law provided some guildlines on selecting values for those parameters, they might not be applied to all cases. What's more, the procedure requires repeatedly running generalized least square regression, which is very complex and computationally expensive.

Yücesan (1993) presents a method based on randomization tests. In this method, the simulation output data are grouped into b batches and batch means are calculated to form a time-series. Then the randomization test is conducted to test the null hypothesis that there is no initialization bias in the batch means, in other words, the batch mean is unchanged in the time-series. At the beginning, the batch means are partitioned into two groups, where the first group consists of the first batch mean and the second group includes the remaining $b - 1$ batch means. The grand means of the two groups are then compared to see if the difference between the two means is significantly different from zero. And a randomization procedure is used here to shuffle the batch means to compute the significance level. The procedure is repeated if the hypothesis is rejected; the first group now includes the first two batch means and the second group consists of the last $b - 2$ batch means. The procedure will be terminated if the test fails to reject the null hypothesis, and the batches in the second group are from steady-state. The advantage of using this method is that there is no assumption about the distribution of the output data. However, the method requires

very large batch size to reduce potential correlations in the data and the shuffling of the batch means could be computationally expensive.

2.3.4 Initialization Bias Test Methods

Methods in this category is not designed to detect the initial transient period directly. Instead, they are often used to test whether the warm-up period has been detected.

Schruben (1982) introduces a maximum test method to test the maximum difference between the mean of the entire output time-series data and the mean of the first k observations. An F-test is then conducted for the bias statistics. Schruben et al. (1983) propose an optimal test method that follows the same principle except that a t-test is used. Although these bias tests perform well when a large initialization bias presents, the choice of the sample size might affect the power of the tests. Goldsman et al. (1994) extend the idea of the maximum test and suggest using batch means instead of the original output time-series. They also propose two alternative tests: batch-means test and area test. The batch-means test groups the output into two sets of batches and computes the bias statistics based on the variance of the batch means. The area test differentiates the batch-means test by computing the test statistics based on the area under standardized time-series of the batch means.

2.3.5 Hybird Methods

Hybrid methods usually integrate bias tests with graphical methods or heuristic methods to detect the truncation point for the warm-up period. Pawlikowski (1990) describes a sequential procedure based on the optimal test. In the procedure, a graphical or heuristic method is first used to estimate the initial truncation point. The optimal test is then used for the first batch of the truncated time-series to test the null hypothesis that there is no initialization bias. If the test rejects the null hypothesis, new simulation output data are collected and the procedure will be repeated until the

test fails to reject the null hypothesis. Although this procedure takes advantages of the initialization bias tests and is quite efficient by sequentially screening the data, the performance requires the estimation of several parameters such as the variance, the initial truncation point and the number of observations included in the test. Its performance also heavily relies on the graphic or heuristic method used to detect the truncation point at each iteration.

Chapter 3

Simulation-based Optimization over Discrete Sets with Noisy Constraints

3.1. Overview

We consider the optimization problem of the form:

$$\begin{aligned} \min_{\theta \in C \cap \mathbb{Z}^d} \quad & f^0(\theta) \\ \text{s/t} \quad & f^i(\theta) \leq 0, \quad 1 \leq i \leq r, \end{aligned} \tag{3.1}$$

where C is a nonempty closed convex subset of $\mathbb{R}^d$, and $f^i : \mathbb{Z}^d \to \mathbb{R}$ $(0 \leq i \leq r)$ has no analytic form and thus must be computed only through simulation at each θ in $\mathbb{Z}^d$. Since we can only obtain noisy simulation observations of constraint functions, it is very difficult to know if $f^i(\theta) \leq 0 (0 \leq i \leq r)$ for sure. Hence, we consider converting the original constrained problem into an unconstrained one to overcome this issue. In particular, we observe that the minimizer of the constrained problem (3.1) can be found by finding the saddle point of the corresponding Lagrangian $L : \mathbb{Z}^d \times \mathbb{R}^r_+ \to \mathbb{R}$ defined by $L(\theta, \lambda) = f^0(\theta) + \sum_{i=1}^r \lambda^i f^i(\theta)$ for $\theta \in \mathbb{Z}^d$ and $\lambda = (\lambda^1, \ldots, \lambda^r) \in \mathbb{R}^r_+$; i.e., if L has a saddle point (θ_*, λ_*) satisfying $L(\theta_*, \lambda) \leq L(\theta_*, \lambda_*) \leq L(\theta, \lambda_*)$ for all $\theta \in C \cap \mathbb{Z}^d$ and $\lambda \in \mathbb{R}^r_+$, then θ_* is a minimizer of f^0 subject to the constraints $f^i \leq 0$

for $1 \leq i \leq r$. From this observation, we reformulate (3.1) as

$$\max_{\lambda \in \mathbb{R}_+^r} \min_{\theta \in C \cap \mathbb{Z}^d} L(\theta, \lambda) \tag{3.2}$$

and propose a gradient-based method to solve (3.2). In particular, we wish to apply stochastic approximation to L (and hence update θ in the steepest descent direction and λ in the steepest ascent direction in each iteration) in search of the saddle point of L. Stochastic approximation requires gradient estimates of L in each iteration, but θ is integer-valued; therefore, the gradient of L with respect to θ cannot be defined in a traditional way. To overcome this obstacle, we extend L from $\mathbb{Z}^d \times \mathbb{Z}_+^r$ to $\mathbb{R}^d \times \mathbb{R}_+^r$ by extending f^i $(0 \leq i \leq r)$ from a discrete domain to a continuous domain and using the extended f^is to construct the extended L, and compute the gradient of the extended L in the usual way. The gradient of the extended L is then used in each iteration of stochastic approximation in search of the solution to (3.2). We then propose the Theorem 1 of this dissertation proposal which states that this procedure is convergent to the optimal solution of the original problem (3.1) almost surely (a.s.) under suitable conditions.

3.1.1 Lagrangian Function versus Penalty Function

Our proposed method utilizes Lagrangian functions rather than penalty functions because the Lagrangian method has certain numerical advantages over the penalty function type methods in the deterministic optimization context. The difference between the Lagrangian function method and penalty function method can be summarized as follows:

1. The penalty function method typically involves solving a sequence of nonlinear

optimization problems (p. 479 of Bazaraa et al. (2006)). Each of these optimization problems must be solved through numerical procedures. Thus, the computational burden of solving these optimization problems can be significant.

2. When solving the optimization problems in 1, the problems can be ill-conditioned for large values of the penalty parameter (large values of the penalty parameter are required to guarantee convergence to the optimal solution). Thus, they can result in undesirable solutions (p. 481 of Bazaraa et al. (2006)).

We illustrate the difference between the Lagrangian function method and penalty function method more clearly by solving the following two-dimensional problem:

$$\begin{aligned} \min_{\theta=(\theta^1,\theta^2)} \quad & f^0(\theta) = (\theta^1 - 2)^2 + (\theta^2 - 4)^2 \\ \text{s/t} \quad & f^1(\theta) = (\theta^1 - 1)^2 + (\theta^2 - 1)^2 - 5 \leq 0. \end{aligned} \tag{3.3}$$

In the Lagrangian function method, we solve (3.3) by finding the saddle point of the corresponding Lagrangian function $L(\theta, \lambda) = f^0(\theta) + \lambda f^1(\theta)$ for $\lambda \geq 0$. We use a gradient-based method and update θ and λ iteratively using the following recursion:

$$\begin{aligned} \theta_{n+1} &= \theta_n - a_n \nabla L_\theta \left(\theta_n, \lambda_n\right), \\ \lambda_{n+1} &= \max\left(0, \lambda_n + a_n \nabla L_\lambda \left(\theta_n, \lambda_n\right)\right), \end{aligned}$$

where $\nabla L_\theta \left(\theta_n, \lambda_n\right)$ and $\nabla L_\lambda \left(\theta_n, \lambda_n\right)$ are the gradients of the Lagrangian function L with respect to θ and λ at (θ_n, λ_n), respectively, and a_n is a decreasing sequence of positive real numbers (Zangwill, 1969). Figure 3.1 shows the performance of the Lagrangian function method within 5000 iterations when $a_n = 0.6/(n+1)$ for $n \geq 0$, $\lambda_0 = 0$, and $\theta_0 = (0, 0)$. In Figure 3.1, we can observe that the Lagrangian function method does not stay in the feasible region all the time, but it converges to a feasible

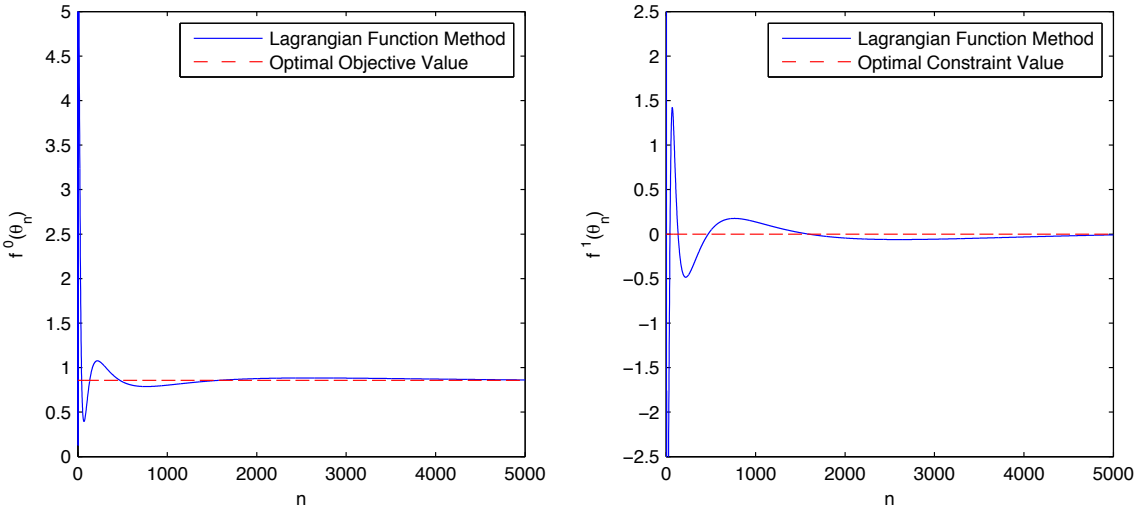

Figure 3.1: The graph of $f^0(\theta_n)$ versus n (left) and graph of $f^1(\theta_n)$ versus n (right). At the optimal solution θ_*, $f^0(\theta_*) = 0.8579$ and $f^1(\theta_*) = 0$.

solution for sufficiently large n by adjusting the λ values adaptively. When θ_n enters an infeasible region, the Lagrangian function method updates λ in the steepest ascent direction to increase the penalties on the violated constraints and changes the search direction of θ to force the constraints into satisfaction. This procedure guarantees $\{\theta_n, \lambda_n\}$ converge to the saddle point of the Lagrangian function, and hence, θ_n stays in the feasible region for n sufficiently large.

In the penalty function method, we convert (3.3) into an unconstrained problem defined by

$$\min_{\theta=(\theta^1,\theta^2)} \qquad L_p(\theta) = f^0(\theta) + \frac{b}{2}\max(0, f^1(\theta))^2, \tag{3.4}$$

where b is a positive real number, which is typically referred to as the penalty value. Some unconstrained optimization techniques, such as line search methods, steepest decent methods, and the Newton method can be applied to solve the penalty problem (3.4). To make the solution of the penalty problem (3.4) arbitrarily close to the

optimal solution of original problem (3.3), b should be sufficiently large. However, with a large value of b, (3.4) may be ill-conditioned. This is because with a large value of b, more emphasis is placed on the penalty part of $L_p(\theta)$, $\frac{b}{2}\max(0, f^1(\theta))^2$. As a result, most unconstrained optimization techniques will force the constraints to be satisfied by moving toward a feasible point, which might be far from the optimal solution (Bazaraa et al., 2006). Figure 3.2 shows the performance of the penalty function method when b takes different values. In this implementation, a steepest decent method is used to solve each unconstrained problem starting from $\theta = (10, 10)$. In Figure 3.2, we can observe that for a large value of b ($b > 10^3$), the solution to

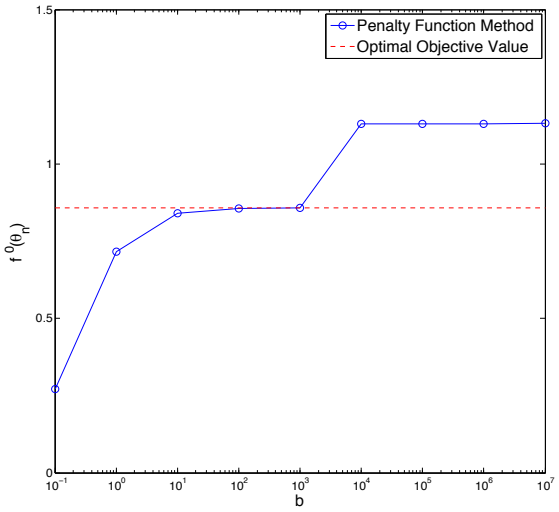

Figure 3.2: The plot of $f^0(\theta_n)$ versus b. At the optimal solution θ_*, $f^0(\theta_*) = 0.8579$.

(3.3) deviates from the optimal solution because of ill-conditioning.

We present the effectiveness of our proposed method numerically in an illustrative example. The proposed method displays good performance when compared with alternative approaches.

The main advantages of the proposed method can be summarized as follows: (1) it is designed to handle stochastic constraints when the number of feasible solutions is large or infinite, (2) it is shown to be convergent to the optimal solution a.s. under

certain technical conditions, and (3) it shows promising numerical performance in various examples.

This chapter is organized as follows. In Section 3.2, we introduce some definitions. Section 3.3 describes our proposed method formally and states the main theorem (Theorem 1) of this chapter. Section 3.4 presents the numerical results and compares the proposed method with some other methods in the literature. Section 3.5 provides the proof for the main theorem.

3.2. Definitions

In this section, we introduce some definitions that will be used throughout this chapter. For a positive integer m, $\mathbb{Z}^m$, $\mathbb{R}^m$, and $\mathbb{R}^m_+$ denote the set of m–dimensional integer vectors, the set of m–dimensional real vectors, and the set of m–dimensional nonnegative real vectors, respectively. We view vectors as columns and write x^T to denote the transpose of a vector $x \in \mathbb{R}^m$. For $x \in \mathbb{R}^m$, we write its jth component as x^j, so $x = (x^1, \ldots, x^m)$. By $\|x\|$, we denote $((x^1)^2 + \cdots + (x^m)^2)^{1/2}$. For a subset I of $\{1, \ldots, m\}$, χ_I is an m–dimensional vector whose jth entry is 1 when j belongs to I and 0 otherwise ($1 \leq j \leq m$). We denote $\chi_{\{j\}}$ by e_j ($1 \leq j \leq m$).

For $x \in \mathbb{R}$, $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the largest integer less than or equal to x and the smallest integer greater than or equal to x, respectively. For $x = (x^1, \ldots, x^m) \in \mathbb{R}^m$, $\lfloor x \rfloor$ and $\lceil x \rceil$ denote $(\lfloor x^1 \rfloor, \ldots, \lfloor x^m \rfloor)$ and $(\lceil x^1 \rceil, \ldots, \lceil x^m \rceil)$, respectively. We denote the closest integer point to $x \in \mathbb{R}^m$ by $[x]$.

For a function $g : \mathbb{R}^m \to \mathbb{R}$, a vector $\xi \in \mathbb{R}^m$ is said to be a subgradient of g at $x \in \mathbb{R}^m$ if $g(y) \geq g(x) + \xi^T(y - x)$ for all $y \in \mathbb{R}^m$. If g is convex, then a subgradient of g at x exists at every $x \in \mathbb{R}^m$ (see Theorem 23.4 in p. 217 of Rockafellar (1970)).

For a function $g : \mathbb{R}^m \to \mathbb{R}$, the partial derivative of g with respect to the jth

component at $x \in \mathbb{R}^m$ is denoted by $\partial g(x)/\partial x^j$ for $1 \leq j \leq m$ if the partial derivative exists.

For $x \in \mathbb{R}$, $\max(0, x)$ is x if $x \geq 0$, and zero otherwise.

3.3. Problem Formulation

3.3.1 General Approach

We consider the following problem

$$\begin{aligned} \min_{\theta \in C \cap \mathbb{Z}^d} \quad & f^0(\theta) \\ \text{s/t} \quad & f^i(\theta) \leq 0, \quad 1 \leq i \leq r, \end{aligned} \tag{3.5}$$

where C is a nonempty closed convex subset of $\mathbb{R}^d$ and we can observe $f^i : \mathbb{Z}^d \to \mathbb{R}$ $(0 \leq i \leq r)$ via simulation at each $\theta \in \mathbb{Z}^d$.

To convert (3.5) into an unconstrained problem, we consider the Lagrangian $L : \mathbb{Z}^d \times \mathbb{R}^r_+ \to \mathbb{R}$ defined as follows:

$$L(\theta, \lambda) = f^0(\theta) + \sum_{i=1}^{r} \lambda^i f^i(\theta)$$

for $\theta \in \mathbb{Z}^d$ and $\lambda = (\lambda^1, \ldots, \lambda^r) \in \mathbb{R}^r_+$. We observe that if $\theta_* \in C \cap \mathbb{Z}^d$ and $\lambda_* = (\lambda^1_*, \ldots, \lambda^r_*) \in \mathbb{R}^r_+$ satisfy

i) θ_* minimizes $L(\theta, \lambda_*)$ over $\theta \in C \cap \mathbb{Z}^d$, and

ii) For $1 \leq i \leq r$, $\lambda^i_* > 0$ implies $f^i(\theta_*) = 0$ and $\lambda^i_* = 0$ implies $f^i(\theta_*) \leq 0$,

then θ_* is an optimal solution to (3.5). To see why this is true, we note that i) and ii) imply that $\theta_* \in \mathcal{F}$, where $\mathcal{F} \triangleq \left\{\theta \in C \cap \mathbb{Z}^d : f^i(\theta) \leq 0 \text{ for } 1 \leq i \leq r\right\}$. In addition, for any $\theta \in \mathcal{F}$, we have $f^0(\theta_*) + \sum_{i=1}^r \lambda^i_* f^i(\theta_*) \leq f^0(\theta) + \sum_{i=1}^r \lambda^i_* f^i(\theta)$, and hence, $f^0(\theta_*) \leq f^0(\theta) + \sum_{i=1}^r \lambda^i_* \left(f^i(\theta) - f^i(\theta_*)\right) \leq f^0(\theta)$.

Furthermore, it can be easily seen that i) and ii) are equivalent to the condition

$$L(\theta_*, \lambda) \leq L(\theta_*, \lambda_*) \leq L(\theta, \lambda_*)$$

for all $\theta \in C \cap \mathbb{Z}^d$ and $\lambda \in \mathbb{R}^r_+$; i.e., (θ_*, λ_*) is a saddle-point of L.

Therefore, it is reasonable to attempt to find the optimal solution to (3.5) by solving

$$\max_{\lambda \in \mathbb{R}^m_+} \min_{\theta \in C \cap \mathbb{Z}^d} L(\theta, \lambda). \tag{3.6}$$

In our proposed method, we search for the solution to (3.6) by updating θ and λ iteratively using a gradient-based method. One obstacle to this approach is that L has a discrete input variable θ, so it is impossible to define the gradient of L with respect to θ in a traditional way. To overcome this, we extend L from $\mathbb{Z}^d \times \mathbb{R}^r_+$ to $\mathbb{R}^d \times \mathbb{R}^r_+$ and compute the gradient of the extended L in the usual way. In order to extend L, we extend f^i $(0 \leq i \leq r)$ from $\mathbb{Z}^d$ to $\mathbb{R}^d$ and use the extended functions to extend L. In particular, we denote the extension of f^i by $\widehat{f^i}$ $(\widehat{f^i} : \mathbb{R}^d \to \mathbb{R})$ for $0 \leq i \leq r$ and define $\widehat{L} : \mathbb{R}^d \times \mathbb{R}^r_+ \to \mathbb{R}$ by

$$\widehat{L}(\theta, \lambda) = \widehat{f^0}(\theta) + \sum_{i=1}^{r} \lambda^i \widehat{f^i}(\theta)$$

for $\theta \in \mathbb{R}^d$ and $\lambda = (\lambda^1, \ldots, \lambda^r) \in \mathbb{R}^r_+$.

Our proposed method then solves

$$\max_{\lambda \in \mathbb{R}^r_+} \min_{\theta \in C \cap \mathbb{R}^d} \widehat{L}(\theta, \lambda) \tag{3.7}$$

in the hope of solving (3.6). We observe that if $(\widehat{\theta}_*, \widehat{\lambda}_*)$ solves (3.7), then $\widehat{\theta}_*$ solves the following problem:

$$\min_{\theta \in C \cap \mathbb{R}^d} \quad \widehat{f^0}(\theta) \tag{3.8}$$

$$\text{s/t} \qquad \widehat{f}^i(\theta) \leq 0, \quad 1 \leq i \leq r$$

(see Theorem 2.18 in p. 48 of Zangwill (1969)), which can be viewed as a relaxed version of (3.5). Thus the remaining question is the relationship between the solution to (3.8) and the solution to (3.5). The following proposition confirms their relationship in the case where the solution to (3.8) is an integer point.

Proposition 1. *Suppose that there exists a solution to (3.8), say* $\widehat{\theta}_*$*, and that* $\widehat{\theta}_* \in \mathbb{Z}^d$*. Then* $\widehat{\theta}_*$ *is a solution to (3.5). Therefore, if* $(\widehat{\theta}_*, \widehat{\lambda}_*)$ *is a saddle point of* $\widehat{L}$*, i.e.,*

$$\widehat{L}(\widehat{\theta}_*, \lambda) \leq \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \leq \widehat{L}(\theta, \widehat{\lambda}_*)$$

for all $\theta \in C \cap \mathbb{R}^d$ *and* $\lambda \in \mathbb{R}^r_+$*, and* $\widehat{\theta}_*$ *is an integer point, then* $\widehat{\theta}_*$ *is an optimal solution to (3.5).*

Proof. Let $\widehat{\theta}_*$ be a solution to (3.8) and assume that $\widehat{\theta}_* \in \mathbb{Z}^d$. Then for any $\theta \in C \cap \mathbb{Z}^d$ satisfying $\widehat{f}^i(\theta) = f^i(\theta) \leq 0$ $(1 \leq i \leq r)$, we have $\widehat{f}^0(\widehat{\theta}_*) \leq \widehat{f}^0(\theta)$. Since $\widehat{\theta}_*$ and θ are integer points, we have $f^0(\widehat{\theta}_*) \leq f^0(\theta)$. So $\widehat{\theta}_*$ is an optimal solution to (3.5).

To prove the last part, we observe that if $\widehat{\theta}_* \in C \cap \mathbb{Z}^d$ and $\widehat{\lambda}_* = (\widehat{\lambda}^1_*, \ldots, \widehat{\lambda}^r_*) \in \mathbb{R}^r_+$ satisfy

i) $\widehat{\theta}_*$ minimizes $\widehat{L}(\theta, \widehat{\lambda}_*)$ over $\theta \in C \cap \mathbb{R}^d$, and

ii) for $1 \leq i \leq r$, $\widehat{\lambda}^i_* > 0$ implies $\widehat{f}^i(\widehat{\theta}_*) = 0$ and $\widehat{\lambda}^i_* = 0$ implies $\widehat{f}^i(\widehat{\theta}_*) \leq 0$,

then $\widehat{\theta}_*$ is an optimal solution to (3.5).

To see why this is true, we note that ii) implies that $\widehat{\theta}_* \in \mathcal{F} \triangleq \{\theta \in C \cap \mathbb{Z}^d : f^i(\theta) \leq 0 \text{ for } 1 \leq i \leq r\}$. In addition, for any $\theta \in \mathcal{F}$, from i), we have

$$f^0(\widehat{\theta}_*) + \sum_{i=1}^r \widehat{\lambda}^i_* f^i(\widehat{\theta}_*) \leq f^0(\theta) + \sum_{i=1}^r \widehat{\lambda}^i_* f^i(\theta),$$

and hence,

$$\begin{aligned} f^0(\widehat{\theta}_*) &\leq f^0(\theta) + \sum_{i=1}^{r} \widehat{\lambda}_*^i \left(f^i(\theta) - f^i(\widehat{\theta}_*) \right) \\ &\leq f^0(\theta). \end{aligned}$$

Therefore, $\widehat{\theta}_*$ is an optimal solution to (3.5).

It remains to show that i) and ii) are implied by the condition:

$$\widehat{L}(\widehat{\theta}_*, \lambda) \leq \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \leq \widehat{L}(\theta, \widehat{\lambda}_*) \tag{3.9}$$

for all $\theta \in C \cap \mathbb{R}^d$ and $\lambda \in \mathbb{R}^r_+$.

Suppose that (3.9) is true. From $\widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \leq \widehat{L}(\theta, \widehat{\lambda}_*)$, $\widehat{\theta}_*$ minimizes $\widehat{L}(\theta, \widehat{\lambda}_*)$ over $\theta \in C \cap \mathbb{R}^d$. Hence, i) follows.

On the other hand, from $\widehat{L}(\widehat{\theta}_*, \lambda) \leq \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*)$, we have

$$\widehat{f}^0(\widehat{\theta}_*) + \sum_{i=1}^{r} \lambda^i \widehat{f}^i(\widehat{\theta}_*) \leq \widehat{f}^0(\widehat{\theta}_*) + \sum_{i=1}^{r} \widehat{\lambda}_*^i \widehat{f}^i(\widehat{\theta}_*)$$

for all $\lambda = (\lambda^1, \ldots, \lambda^r) \in \mathbb{R}^r_+$. Since $\widehat{\lambda}_* \in \mathbb{R}^r_+$ and this inequality holds for all $\lambda \in \mathbb{R}^r_+$, we must have $\widehat{f}^i(\widehat{\theta}_*) \leq 0$ for all $1 \leq i \leq r$ because if $\widehat{f}^j(\widehat{\theta}_*) > 0$ for some j, then the inequality is violated by $\lambda = (0, \ldots, 0, \lambda^j, 0, \ldots, 0)$ with $\lambda^j > 0$ sufficiently large. This implies

$$\widehat{f}^i(\widehat{\theta}_*) \leq 0$$

for all $1 \leq i \leq r$ and

$$\sum_{i=1}^{r} \widehat{\lambda}_*^i \widehat{f}^i(\widehat{\theta}_*) \leq 0.$$

In addition, if we let $\lambda = (0, \ldots, 0)$, we obtain

$$\sum_{i=1}^{r} \widehat{\lambda}_*^i \widehat{f}^i(\widehat{\theta}_*) \geq 0.$$

Thus, we must have $\sum_{i=1}^{r} \widehat{\lambda}_*^i \widehat{f}^i(\widehat{\theta}_*) = 0$. Since $\widehat{f}^i(\widehat{\theta}_*) \leq 0$ and $\widehat{\lambda}_*^i \geq 0$ for all $1 \leq i \leq r$, we must have $\widehat{\lambda}_*^i \widehat{f}^i(\widehat{\theta}_*) = 0$ for $1 \leq i \leq r$, and hence, ii) follows.

□

The framework of the proposed method then can be summarized as follows:

1. Extend L from a discrete domain to a continuous domain and obtain the extended function $\widehat{L}$.

2. Obtain subgradients of $\widehat{L}$ with respect to θ and λ, respectively.

3. Apply stochastic approximation to $\widehat{L}$ to find the saddle point of $\widehat{L}$.

Given the relationship between the saddle point of $\widehat{L}$ and the solution to (3.5), the proposed method applies stochastic approximation to $\widehat{L}$ in order to search for the saddle point. Denoting the nth estimator of the saddle point of $\widehat{L}$ by (θ_n, λ_n), we update θ_n and λ_n as follows. Given $(\theta_1, \lambda_1), \ldots, (\theta_n, \lambda_n)$, we observe a quantity $-D_n(\theta_n, \lambda_n)$ that guides us towards the steepest descent direction in θ ($-D_n(\theta_n, \lambda_n)$ can be interpreted as the negative of the derivative of $\widehat{L}$ with respect to θ if $\widehat{L}$ is differentiable in θ or the negative of the subgradient of $\widehat{L}$ in θ if $\widehat{L}$ is convex in θ). We then update θ_n by the recursion

$$\theta_{n+1} = \Pi_C \left(\theta_n - c_n D_n(\theta_n, \lambda_n) \right),$$

where $\Pi_C(\theta)$ is the closest point in C to $\theta \in \mathbb{R}^d$ with respect to the norm $\|\cdot\|$. $(c_n : n \geq 1)$ is a sequence of positive real numbers satisfying the conditions $\sum_{n=1}^{\infty} c_n = \infty$ and $\sum_{n=1}^{\infty} c_n^2 \leq \infty$. The specific form of this sequence will be provided in the numerical experiments part.

Using the fact that

$$\partial \widehat{L}(\theta, \lambda)/\partial \lambda^i \;\; = \;\; \widehat{f}^i(\theta)$$

for $1 \leq i \leq r$ provided that the partial derivative exists, we update λ_n by the recursion

$$\lambda_{n+1}^i = \max\left(0, \lambda_n^i + c_n F^i(\theta_n)\right)$$

for $1 \leq i \leq r$, where $F^i(\theta_n)$ is an observation of $\widehat{f}^i$ at θ_n.

Finally, we assume that there exists a known bound K for a saddle point of $\widehat{L}$; i.e., there exists a positive constant K such that $|\widehat{\theta}_*^i| \leq K$ for $1 \leq i \leq d$ and $|\widehat{\lambda}_*^i| \leq K$ for $1 \leq i \leq r$, where $(\widehat{\theta}_*, \widehat{\lambda}_*)$ is a saddle point of $\widehat{L}$. With this additional information, we project (θ_n, λ_n) onto $\mathcal{B} \triangleq \{(\theta, \lambda) \in \mathbb{R}^d \times \mathbb{R}^r : |\theta^i| \leq K \text{ for } 1 \leq i \leq d, |\lambda^i| \leq K \text{ for } 1 \leq i \leq r\}$ and the projected point is $(\theta_{n+1}, \lambda_{n+1})$.

Our proposed method takes the following form in general:

Algorithm 1: General Form of the Proposed Method

Step 0. Initialize: Select a starting point $(\theta_0, \lambda_0) \in \mathbb{Z}^d \times \mathbb{R}^r$. Set $n = 0$.

Step 1. Update θ_n and λ_n: Generate observations $D_n(\theta_n, \lambda_n)$ and $F^i(\theta_n)$ for $1 \leq i \leq r$ and set

$$\begin{aligned} \overline{\theta}_{n+1} &= \Pi_C\left(\theta_n - c_n D_n(\theta_n, \lambda_n)\right), \\ \overline{\lambda}_{n+1}^i &= \max\left(0, \lambda_n^i + c_n F^i(\theta_n)\right) \end{aligned}$$

for $1 \leq i \leq r$.

Step 2. Project $(\overline{\theta}_{n+1}, \overline{\lambda}_{n+1})$ onto $\mathcal{B}$ and set the projected point equal to $(\theta_{n+1}, \lambda_{n+1})$.

Step 3. Let $n = n + 1$ and go to Step 1.

To investigate the asymptotic behavior of $((\theta_n, \lambda_n) : n \geq 1)$, we focus on the case where $\widehat{f^i}$ $(0 \leq i \leq r)$ is convex. In this case, differentiability of $\widehat{f^i}$ is not necessary; only D_n needs to be an unbiased estimate of a subgradient of $\widehat{L}$ in θ and F^i needs to be an unbiased estimate of $\widehat{f^i}$ for $1 \leq i \leq r$. In particular, we require:

A1. $(c_n : n \geq 1)$ is a sequence of positive numbers satisfying $\sum_{n=1}^{\infty} c_n = \infty$ and $\sum_{n=1}^{\infty} c_n^2 \leq \infty$.

A2. $\widehat{f^0}$ is strictly convex and $\widehat{f^i}$ $(1 \leq i \leq r)$ is convex.

A3. There exists $\eta \in C$ such that $\widehat{f^i}(\eta) < 0$ for $1 \leq i \leq r$. The optimal value of (3.8) is finite.

A4. $\mathbb{E}\left[D_n(\theta_n, \lambda_n)|\mathcal{F}_n\right]$ is a subgradient of $\widehat{L}$ at (θ_n, λ_n) as a function of θ; i.e.,

$$\widehat{L}(\theta, \lambda_n) \;\; \geq \;\; \widehat{L}(\theta_n, \lambda_n) + \mathbb{E}\left[D_n(\theta_n, \lambda_n)|\mathcal{F}_n\right]^T (\theta - \theta_n) \tag{3.10}$$

for all $\theta \in \mathbb{R}^d$, where $\mathcal{F}_n$ is the σ–field generated by $(\theta_1, \lambda_1), \ldots, (\theta_n, \lambda_n)$. In addition, we assume

$$\mathbb{E}\left[F^i(\theta_n)|\mathcal{F}_n\right] \;\; = \;\; \widehat{f^i}(\theta_n) \tag{3.11}$$

for $1 \leq i \leq r$ and $n \geq 1$,

$$\mathbb{E}\left[\|D_n(\theta_n, \lambda_n) - \mathbb{E}\left[D_n(\theta_n, \lambda_n)|\mathcal{F}_n\right]\|^2|\mathcal{F}_n\right] \;\; < \;\; \sigma^2, \tag{3.12}$$

for $n \geq 1$, and

$$\mathbb{E}\left[(F^i(\theta_n) - \mathbb{E}\left[F^i(\theta_n)|\mathcal{F}_n\right])^2|\mathcal{F}_n\right] \;\; < \;\; \sigma^2 \tag{3.13}$$

for $1 \leq i \leq r$ and $n \geq 1$ for some positive constant σ^2.

Theorem 1. *Under A1–A4, i) there exists a saddle point $(\widehat{\theta}_*, \widehat{\lambda}_*)$ of $\widehat{L}$, ii) $\widehat{\theta}_*$ is unique, and iii) $\theta_n \to \widehat{\theta}_*$ a.s. as $n \to \infty$. By Proposition 1, if $\widehat{\theta}_*$ is an integer point, then $\widehat{\theta}_*$ is a solution to (3.5).*

3.3.2 Extension via Piecewise Linear Interpolation

The continuous extensions $\widehat{f^i}$ $(0 \leq i \leq r)$ introduced in Section 3.3.1 can be chosen arbitrarily. However to make our procedure more concrete, we introduce one possible way of extending functions from $\mathbb{Z}^d$ to $\mathbb{R}^d$. In particular, we consider extending a function $h : \mathbb{Z}^d \to \mathbb{R}$ via the piecewise linear interpolation over a particular partition of $\mathbb{R}^d$ as follows. For $\theta \in \mathbb{R}^d$, we let $p = \lfloor \theta \rfloor$ and $q = (q^1, \ldots, q^d) = \theta - p$. σ is the permutation of $(1, \ldots, d)$ such that $\sigma(j)$ is the index of the jth largest of $q^1, \ldots, q^d$ (if $q^{\sigma(j)} = q^{\sigma(k)}$ for some j and k, then let $\sigma(j) > \sigma(k)$ when $j > k$). We set $U_0 = \emptyset$ and $U_j = \{\sigma(1), \ldots, \sigma(j)\}$ for $1 \leq j \leq d$. We define $\widetilde{h} : \mathbb{R}^d \to \mathbb{R}$ by

$$\begin{aligned} \widetilde{h}(\theta) &= \left(1 - q^{\sigma(1)}\right) h(p) + \left(q^{\sigma(1)} - q^{\sigma(2)}\right) h(p + \chi_{U_1}) \\ &\quad + \cdots + \left(q^{\sigma(d-1)} - q^{\sigma(d)}\right) h\left(p + \chi_{U_{d-1}}\right) + q^{\sigma(d)} h\left(p + \chi_{U_d}\right). \end{aligned} \tag{3.14}$$

By construction, $\widetilde{h}(\theta) = h(\theta)$ for $\theta \in \mathbb{Z}^d$, so $\widetilde{h}$ is a continuous extension of h over $\mathbb{R}^d$.

Even though $\widetilde{h}$ is not differentiable at some points in $\mathbb{R}^d$, a subgradient of $\widetilde{h}$ can be easily computed when $\widetilde{h}$ is convex. We define $\varphi\widetilde{h}(\theta) = (\varphi\widetilde{h}^j(\theta) : j = 1, \ldots, d)$ by

$$\varphi\widetilde{h}^j(\theta) = \widetilde{h}(p + \chi_{U_k}) - \widetilde{h}(p + \chi_{U_{k-1}}) \tag{3.15}$$

for $\theta \in \mathbb{R}^d$, where $q^j = q^{\sigma(k)}$.

The following propositions prove that $\varphi\widetilde{h}(\theta)$ is a subgradient of h at $\theta \in \mathbb{R}^d$ when $\widetilde{h}$ is convex.

Proposition 2. *Let $h : \mathbb{Z}^d \to \mathbb{R}$ be given and $\widetilde{h} : \mathbb{R}^d \to \mathbb{R}$ be defined by (3.14). For any $\theta \in \mathbb{R}^d$ and $\delta > 0$, i) there exists $\theta_\delta \in \mathbb{R}^d$ such that $\|\theta - \theta_\delta\| \le \delta$, ii) $\widetilde{h}$ is differentiable at θ_δ, iii) $\varphi\widetilde{h}(\theta) = \varphi\widetilde{h}(\theta_\delta)$, and iv) $\varphi\widetilde{h}(\theta_\delta)$ is the gradient of $\widetilde{h}$ at θ_δ.*

Proof. Let $\theta \in \mathbb{R}^d$ and $\delta > 0$ be given. Let $p = \lfloor \theta \rfloor$ and $q = \theta - p$. First we consider the case where $q = (q^1, \ldots, q^d)$ has d distinct components and $q^j \neq 0$ for $1 \le j \le d$. In this case, we let $\theta_\delta = \theta$. Note $q^{\sigma(1)} > \cdots > q^{\sigma(d)} > 0$ and by the representation (3.14), there exists a neighborhood of θ where $\widetilde{h}$ is linear. Hence in that neighborhood, $\widetilde{h}$ is differentiable. (In particular, $\widetilde{h}$ is differentiable at $\theta = \theta_\delta$.) To prove that $\varphi\widetilde{h}(\theta)$ is the gradient of $\widetilde{h}$ at θ, it suffices to prove that $\varphi\widetilde{h}_j(\theta)$ is the right derivative of $\widetilde{h}$ at θ in the jth component. For $j \in \{1, \ldots, d\}$, let $q^{\sigma(k)} = q^j$. The right derivative of $\widetilde{h}$ at θ in the jth component is

$$
\begin{aligned}
&\lim_{\gamma \downarrow 0} \left[\widetilde{h}(q^1, \ldots, q^j + \gamma, \ldots, q^d) - \widetilde{h}(q^1, \ldots, q^d)\right] / \gamma \\
&\quad = \lim_{h \downarrow 0} (1/\gamma) \left[\left(\left(1 - q^{\sigma(1)}\right) h(p) + \cdots + \left(q^{\sigma(k-1)} - (q^{\sigma(k)} + \gamma)\right) h(p + \chi_{U_{k-1}}) \right.\right. \\
&\qquad + \left((q^{\sigma(k)} + \gamma) - q^{\sigma(k+1)}\right) h(p + \chi_{U_k}) + \cdots + q^{\sigma(d)} h\left(p + \chi_{U_d}\right) \Big) \\
&\qquad - \left(\left(1 - q^{\sigma(1)}\right) h(p) + \cdots + \left(q^{\sigma(k-1)} - q^{\sigma(k)}\right) h(p + \chi_{U_{k-1}}) \right. \\
&\qquad \left.\left. + \left(q^{\sigma(k)} - q^{\sigma(k+1)}\right) h(p + \chi_{U_k}) + \cdots + q^{\sigma(d)} h\left(p + \chi_{U_d}\right) \right) \right] \\
&\quad = \widetilde{h}(p + \chi_{U_k}) - \widetilde{h}(p + \chi_{U_{k-1}}) \\
&\quad = \varphi\widetilde{h}_j(\theta),
\end{aligned}
$$

proving that $\varphi\widetilde{h}_j(\theta)$ is the right derivative of $\widetilde{h}$ at θ in the jth component.

Now we consider the general case. For any $q^{\sigma(1)} \ge \cdots \ge q^{\sigma(d)} \ge 0$, there exists $q_\delta = (q_\delta^1, \ldots, q_\delta^d)$ such that the order among $q_\delta^1, \ldots, q_\delta^d$ is the same as the order among $q^1, \ldots, q^d$, $q_\delta^1, \ldots, q_\delta^d$ are all distinct, $q_\delta^i \neq 0$ for $1 \le i \le d$, and $\|\theta - (p + q_\delta)\| \le \delta$. Let $\theta_\delta = p + q_\delta$. Then $\|\theta - \theta_\delta\| \le \delta$. Since the order among $q_\delta^1, \ldots, q_\delta^d$ is the same as the order among $q^1, \ldots, q^d$, we have $\varphi\widetilde{h}(\theta) = \varphi\widetilde{h}(\theta_\delta)$. From the previous arguments, $\widetilde{h}$ is differentiable at θ_δ and $\varphi\widetilde{h}(\theta_\delta)$ is the gradient of $\widetilde{h}$ at θ_δ. □

Proposition 3. *Let $h : \mathbb{Z}^d \to \mathbb{R}$ be given and $\widetilde{h}$ and $\varphi\widetilde{h}$ be defined by (3.14) and (3.15), respectively. If $\widetilde{h}$ is convex, then $\varphi\widetilde{h}(\theta)$ is a subgradient of $\widetilde{h}$ at $\theta \in \mathbb{R}^d$.*

Proof. First, we prove that the convexity of $\widetilde{h}$ confirms that $\varphi\widetilde{h}(\theta)$ is a subgradient of $\widetilde{h}$ at $\theta \in \mathbb{R}^d$. For any $x, y \in \mathbb{R}^d$, Proposition 2 guarantees the existence of a sequence $(\eta_i : i \geq 1)$ such that

$$\|x - \eta_i\| \leq 1/i, \tag{3.16}$$

$$\varphi\widetilde{h}(\eta_i) \text{ is a gradient of } \widetilde{h} \text{ at } \eta_i, \tag{3.17}$$

$$\varphi\widetilde{h}(\eta_i) = \varphi\widetilde{h}(x) \tag{3.18}$$

for $i \geq 1$. So

$$\begin{aligned}
\widetilde{h}(y) &= \widetilde{h}(\eta_i - \eta_i + y) \\
&\geq \widetilde{h}(\eta_i) - (\eta_i - y)^T \varphi\widetilde{h}(\eta_i) \quad \text{by (3.17) and convexity of } \widetilde{h} \\
&= \widetilde{h}(\eta_i) - (\eta_i - y)^T \varphi\widetilde{h}(x) \quad \text{by (3.18).}
\end{aligned}$$

Letting $i \to \infty$ and using (3.16) and the continuity of $\widetilde{h}$, we get $\widetilde{h}(y) \geq \widetilde{h}(x) + (y - x)^T \varphi\widetilde{h}(x)$. Hence $\varphi\widetilde{h}(x)$ is a subgradient of $\widetilde{h}$ at x. □

In Example 1, we illustrate how to compute $\varphi\widetilde{h}$.

Example 1. *Suppose $d = 3$ and $\theta = (13.2, 9.4, 20.2)$. Then $p = \lfloor\theta\rfloor = (13, 9, 20)$ and $q = (q^1, q^2, q^3) = \theta - p = (0.2, 0.4, 0.2)$. Since $q^2 > q^1 = q^3$, we have $\sigma = (2, 1, 3)$, $U_0 = \emptyset, U_1 = \{2\}, U_2 = \{2, 1\}$, and $U_3 = \{2, 1, 3\}$. Thus*

$$\widetilde{h}(\theta) = (1 - q^2)h(p) + (q^2 - q^1)h(p + \chi_{\{2\}}) + (q^1 - q^3)h(p + \chi_{\{2,1\}}) + q^3 h(p + \chi_{\{2,1,3\}})$$

and

$$\begin{aligned}
\varphi\widetilde{h}^1 &= \widetilde{h}(p+\chi_{\{2,1\}}) - \widetilde{h}(p+\chi_{\{2\}}),\\
\varphi\widetilde{h}^2 &= \widetilde{h}(p+\chi_{\{2\}}) - \widetilde{h}(p),\\
\varphi\widetilde{h}^3 &= \widetilde{h}(p+\chi_{\{2,1,3\}}) - \widetilde{h}(p+\chi_{\{2,1\}}).
\end{aligned}$$

We are now ready to discuss how this strategy of constructing an extension can be adopted in the proposed method. We define the extensions $\widetilde{f}^i : \mathbb{R}^d \to \mathbb{R}$ $(0 \le i \le r)$ via the linear interpolation by

$$\begin{aligned}
\widetilde{f}^i(\theta) &= \left(1 - q^{\sigma(1)}\right) f^i(p) + \left(q^{\sigma(1)} - q^{\sigma(2)}\right) f^i(p+\chi_{U_1})\\
&\quad + \cdots + \left(q^{\sigma(d-1)} - q^{\sigma(d)}\right) f^i\left(p+\chi_{U_{d-1}}\right) + q^{\sigma(d)} f^i\left(p+\chi_{U_d}\right) \qquad (3.19)
\end{aligned}$$

for $\theta \in \mathbb{R}^d$, where p, q, σ, $U_0, \ldots, U_d$ are defined as before. Using these functions, a continuous extension $\widetilde{L} : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}$ of L is defined as

$$\widetilde{L}(\theta,\lambda) = \widetilde{f}^0(\theta) + \sum_{i=1}^{r} \lambda^i \widetilde{f}^i(\theta)$$

for $\theta \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}^r_+$.

We next define $\varphi\widetilde{f}^i(\theta) = (\varphi\widetilde{f}^i_j(\theta) : j = 1, \ldots, d)$ for $0 \le i \le r$ by

$$\varphi\widetilde{f}^i_j(\theta) = \widetilde{f}^i(p+\chi_{U_k}) - \widetilde{f}^i(p+\chi_{U_{k-1}}) \qquad (3.20)$$

for $\theta \in \mathbb{R}^d$, where $q^j = q^{\sigma(k)}$. Propositions 2 and 3 justify our choice of $\varphi\widetilde{f}^i$ as a subgradient of $\widetilde{f}^i$.

With the $\widetilde{f}^i$s as the extended functions, our proposed method proceeds as follows. We denote the nth estimator of the saddle point of $\widetilde{L}$ by (θ_n, λ_n). Given $(\theta_1,\lambda_1), \ldots, (\theta_n,\lambda_n)$, we observe f^i $(0 \le i \le r)$ at $p_n + \chi_{U_k}$ for $0 \le k \le d$, and

presume that

$$Y_n^i(k) \;=\; f^i(p_n + \chi_{U_k}) + \epsilon_n(k), \tag{3.21}$$

where $p_n = \lfloor \theta_n \rfloor$, $q_n = (q_n^1, \ldots, q_n^d) = \theta_n - p_n$, σ_n is the permutation of $(1, \ldots, d)$ such that $\sigma_n(j)$ is the index of the jth largest of $q_n^1, \ldots, q_n^d$ (if $q_n^{\sigma_n(j)} = q_n^{\sigma_n(k)}$ for some j and k, then let $\sigma_n(j) > \sigma_n(k)$ when $j > k$), $U_0 = \emptyset$, $U_k = \{\sigma_n(1), \ldots, \sigma_n(k)\}$ for $1 \le k \le d$, and $(\epsilon_n(k) : 0 \le k \le d, n \ge 1)$ are mean zero random variables. We then update θ_n and λ_n by the recursion

$$\begin{aligned}
\theta_{n+1} &= \Pi_C \left(\theta_n - c_n D_n(\theta_n, \lambda_n)\right), \\
\lambda_{n+1}^i &= \max\left(0, \lambda_n^i + c_n \left((1 - q_n^{\sigma(1)}) Y_n^i(0) + \cdots + q_n^{\sigma(d)} Y_n^i(d)\right)\right)
\end{aligned}$$

for $1 \le i \le r$, where the jth element of $D_n(\theta_n, \lambda_n)$ is $Y_n^0(k) - Y_n^0(k-1) + \sum_{i=1}^r \lambda_n^i (Y_n^i(k) - Y_n^i(k-1))$ with $q_n^j = q_n^{\sigma(k)}$. We then project $(\theta_{n+1}, \lambda_{n+1})$ onto $\mathcal{B}$ and the projected point becomes $(\theta_{n+1}, \lambda_{n+1})$.

We observe that under the assumption that the $\epsilon_n(k)$s are mean zero random variables, (3.20) and (3.21) imply

$$\mathbb{E}\left[D_n(\theta_n, \lambda_n) | (\theta_1, \lambda_1), \ldots (\theta_n, \lambda_n)\right] = \varphi \widetilde{f}^0(\theta_n) + \sum_{i=1}^r \lambda_n^i \varphi \widetilde{f}^i(\theta_n).$$

Below is the proposed method when we adopt the above procedure.

Algorithm 2: Proposed algorithm with Extensions via Linear Interpolation

---

Step 0. Initialize: Select a starting point $(\theta_0, \lambda_0) \in \mathbb{Z}^d \times \mathbb{R}^r$. Set $n = 0$.

Step 1. Update θ_n and λ_n: Generate an observation $Y_n^i(k)$ of f^i at $p_n + \chi_{U_k}$ for $0 \le i \le r$ and $0 \le k \le d$, where p_n and the U_ks are defined as before. Set

$$\begin{aligned} \overline{\theta}_{n+1} &= \Pi_C\left(\theta_n - c_n D_n(\theta_n, \lambda_n)\right), \\ \overline{\lambda}_{n+1}^i &= \max\left(0, \lambda_n^i + c_n\left((1 - q_n^{\sigma(1)})Y_n^i(0) + \cdots + q_n^{\sigma(d)}Y_n^i(d)\right)\right), \end{aligned}$$

where the jth element of $D_n(\theta_n, \lambda_n)$ is $Y_n^0(k) - Y_n^0(k-1) + \sum_{i=1}^r \lambda_n^i(Y_n^i(k) - Y_n^i(k-1))$ with $q_n^j = q_n^{\sigma(k)}$.

Step 2. Project $(\overline{\theta}_{n+1}, \overline{\lambda}_{n+1})$ onto $\mathcal{B}$ and set the projected point equal to $(\theta_{n+1}, \lambda_{n+1})$.

Step 3. Let $n = n + 1$ and go to Step 1.

To analyze the behavior of $((\theta_n, \lambda_n) : n \ge 1)$ generated from Algorithm 2, we shall impose some assumptions. In particular, we require:

A5. $\widetilde{f}^0$ is strictly convex and $\widetilde{f}^i$ $(1 \le i \le r)$ is convex.

A6. There exists $\eta \in \mathbb{R}^d$ such that $\widetilde{f}^i(\eta) < 0$ for $1 \le i \le r$. Let $f_* = \min_{\theta \in C \cap \mathbb{R}^d} \widetilde{f}^0$ subject to $\widetilde{f}^i(\theta) \le 0$ for $1 \le i \le r$ and assume f_* is finite.

A7. The $\epsilon_n(j)$s are random variables satisfying

$$\mathbb{E}\left[\epsilon_n(j) | \mathcal{F}_n\right] = 0$$

and

$$\mathbb{E}\left[\epsilon_n(j)^2 | \mathcal{F}_n\right] < \kappa^2$$

for $0 \le j \le d$, $n \ge 1$, and some positive constant κ^2, where $\mathcal{F}_n$ is the σ–field generated by $(\theta_1, \lambda_1), \ldots, (\theta_n, \lambda_n)$.

From Theorem 1, we have the following theorem.

Theorem 2. *Under A1 and A5–A7, i) there exists a saddle point* $(\widetilde{\theta}_*, \widetilde{\lambda}_*)$ *of* $\widetilde{L}$*, ii)* $\widetilde{\theta}_*$ *is unique, and iii)* $\theta_n \to \widetilde{\theta}_*$ *a.s. as* $n \to \infty$*. By Proposition 1, if* $\widetilde{\theta}_*$ *is an integer point, then* $\widetilde{\theta}_*$ *is the solution to (3.5).*

3.4. Numerical Results

In this section, we investigate the performance of the proposed algorithms in the following four settings: (1) 2-dimensional quadratic problem, (2) inventory control in a periodically–reviewed single–item inventory system, (3) staffing in a call center that handles multiple types of calls while maintaining a satisfactory level of customer service, and (4) staffing in an emergency room. We then compare the proposed method to the methods proposed by Whitney et al. (2001) and Ahmed et al. (1997).

3.4.1 Competing Methods

The method proposed in Whitney et al. (2001) incorporates a penalty function into the objective function. In the nth iteration of this method, we generate $\Delta_n = (\Delta_n^1, \ldots, \Delta_n^d)$, where the Δ_n^is are independent and identically distributed (iid) random variables taking values $+1$ or -1 with equal probability. Denoting the nth estimator of the optimal solution to (3.5) by θ_n, we then generate an observation Y_+^i of f^i at $\theta_n + \Delta_n$, an observation Y_-^i of f^i at $\theta_n - \Delta_n$, and an observation Y^i of f^i for $0 \leq i \leq r$, and update θ_n by the recursion

$$\theta_{n+1} = \Pi_{C \cap \mathcal{B}_\theta} \left(\theta_n - [a_n H_n(\theta_n)] \right),$$

where the jth component of $H_n(\theta_n)$ is

$$(Y_+^0 - Y_-^0 + b_n \sum_{i=1}^{r} \max(0, Y^i)(Y_+^i - Y_-^i))/(2\Delta_n^j)$$

for $1 \leq j \leq d$ and $\Pi_{C \cap \mathcal{B}_\theta}$ is the projection onto the set $C \cap \mathcal{B}_\theta$ with $\mathcal{B}_\theta \triangleq \{\theta \in \mathbb{R}^d : |\theta^j| \leq K \text{ for } 1 \leq j \leq d\}$. $(a_n : n \geq 1)$ and $(b_n : n \geq 1)$ are sequences of positive real numbers.

On the other hand, at the nth iteration of the method proposed by Ahmed et al. (1997), we set the neighborhood $\mathcal{N}(\theta_n)$ of θ_n as

$$\mathcal{N}(\theta_n) \quad = \quad \{\theta \in \mathbb{Z}^d : \|\theta - \theta_n\| = 1\}, \tag{3.22}$$

and choose a candidate for θ_{n+1}, say θ'_n, from $\mathcal{N}(\theta_n)$ with equal probability. We then generate iid observations of f_i $(1 \leq i \leq r)$ at θ'_n and compute the sample mean and standard deviation, say $\widehat{f^i}$ and $\widehat{\sigma}^i$, of those observations. We consider θ'_n feasible if

$$\widehat{f^i} - t_{n-1,1-\alpha}\widehat{\sigma}^i \leq 0$$

for $1 \leq i \leq r$, where $t_{n-1,1-\alpha}$ is the upper $1-\alpha$ critical point for the t distribution with $n-1$ degrees of freedom. If θ'_n is considered feasible by this criterion, we generate an observation Y_n of f^0 at θ_n and an observation Y'_n of f^0 at θ'_n, and accept θ'_n as θ_{n+1} if $Y'_n \leq Y_n$ or $\exp(-(Y'_n - Y_n)/T_f) > U_n$, where T_f is a positive constant and U_n is a random variable uniformly distributed between 0 and 1. If $Y'_n > Y_n$ and $\exp(-(Y'_n - Y_n)/T) \leq U_n$, then θ_n is chosen as θ_{n+1}. If θ'_n is not considered feasible, then θ_n is chosen as θ_{n+1}. We repeat this procedure M times after which T_f is replaced by $T_f R$, where $R < 1$ is a positive constant. The procedure is repeated M times again until T_f is replaced by $T_f R$ again. This process is repeated until a stopping criterion is satisfied.

The subsequent section reports the performances of the proposed method and the two competing methods.

3.4.2 An Illustrative Example

We consider the following 2-dimensional quadratic problem:

$$\begin{aligned} \min_{\theta=(\theta^1,\theta^2)} \quad & f^0(\theta) = \mathbb{E}\left[(\theta^1-10)^2 + (\theta^2-30)^2 + \epsilon_0\right] \\ \text{s/t} \quad & f^1(\theta) = \mathbb{E}\left[(\theta^1)^2 + (\theta^2)^2 + \epsilon_1\right] \leq 0, \end{aligned} \tag{3.23}$$

where $\theta \in \mathbb{Z}^2$. We add iid zero-mean Gaussian noise to both $f^0(\theta)$ and $f^1(\theta)$, i.e., $\epsilon_0 \sim N(0, 2^2)$ and $\epsilon_1 \sim N(0, 5^2)$. We assume that only noisy measurements of the objective function $f^0(\theta)$ and the constraint function $f^1(\theta)$ are available. The deterministic optimal solution θ_* of (3.23) occurs at $(7, 21)$ with $f^0(\theta_*) = 90$ and $f^1(\theta_*) = -10$.

We apply Algorithm 2, and the methods proposed by Whitney et al. (2001) and Ahmed et al. (1997) to find θ_*. Whenever we observe f^0 or f^1 at each point in all three methods, we use the average of 10 iid simulation replications as an observation of f^0 or f^1. The initial solution θ_1 is set as $(0, 0)$ for all the three methods and $\lambda_1 = 0$ is used for Algorithm 2. The other parameters used are $c_n = 0.2/n$, $a_n = 0.2/n$, $b_n = 0.1\log(n^{0.5}), \alpha = 0.95, T_f = 100, R = 0.6, M = 5$.

Denoting the number of simulation runs made at iteration n by t_n and fixing the total number N of simulation runs available, we compute $\theta_{l(N)+1}$ where $l(N)$ is the maximum number of iterations given the N simulation runs available; i.e., $l(N)$ is the largest integer satisfying $t_1 + \cdots + t_{l(N)} \leq N$. Thus, $\theta_{l(N)+1}$ is the best estimate of (θ^1_*, θ^2_*) given the computational budget N. We notice that both f^0 and f^1 can be simultaneously computed in a single simulation run, so $t_n = 30, 30$, and 10 for Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997), respectively. Table 1 reports the averages (Mean) of $\theta_{l(N)+1}$ generated by

Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997) based on 50 independent copies of $\theta_{l(N)}$ for each value of N. To measure how the distribution of $\theta_{l(N)+1}$ is spread out, the average of the sample standard deviation of $\theta^1_{l(N)+1}$ and $\theta^2_{l(N)+1}$ is reported in Table 3.1. In addition, averages of the $f^1(\theta_{l(N)+1})$ values are reported to show that our method converges to a feasible solution for N sufficiently large.

Table 3.1: Averages (Mean) and standard deviation (Std) of $\theta_{l(N)+1}$ and corresponding averages of $f^1(\theta_{l(N)+1})$ generated from three methods applied to the illustrative problem.

	Algorithm 2			Whitney et al. (2001)			Ahmed et al. (1997)		
N	Mean	Std	$f^1(\theta)$	Mean	Std	$f^1(\theta)$	Mean	Std	$f^1(\theta)$
1000	$(7, 22)$	0.07	31.85	$(6, 19)$	2.72	-75.63	$(8, 14)$	2.84	-217.79
2000	$(7, 22)$	0.00	33.05	$(6, 19)$	2.72	-75.50	$(10, 20)$	1.00	-5.14
3000	$(7, 21)$	0.00	-9.27	$(6, 19)$	2.72	-75.39	$(9, 20)$	0.87	-1.91
4000	$(7, 22)$	0.00	33.12	$(6, 19)$	2.72	-75.50	$(9, 20)$	0.87	-2.25
5000	$(7, 21)$	0.00	-9.79	$(6, 19)$	2.72	-75.29	$(10, 20)$	0.87	-1.22
6000	$(7, 21)$	0.00	-9.96	$(6, 19)$	2.72	-75.46	$(10, 20)$	0.87	-1.54
(θ^1_*, θ^2_*)	(7, 21)			(7, 21)			(7, 21)		

From Table 3.1 we can conclude that only Algorithm 2 converges to the optimal solution $(7, 21)$ for N sufficiently large and has the smallest Std value.

Figure 3.3 and Figure 3.4 show the typical behaviour of the objective function values and the constraint function values for the three methods as the number of simulation runs N is increasing. From Figure 3.3 and Figure 3.4 we can see that both Algorithm 2 and the method proposed by Whitney et al. (2001) converge very quickly. But the later method only converges to a non-optimal feasible solution. We notice that the performance of this penalty function method is quite sensitive to the choice of the penalty value b_n. If b_n is too small, the method cannot guarantee to converge to a optimal solution; if b_n is too large, the method becomes ill-conditioned and returns

some unmeaningful solutions. Even though (3.23) is a simple optimization problem, the penalty function method does not perform very well.

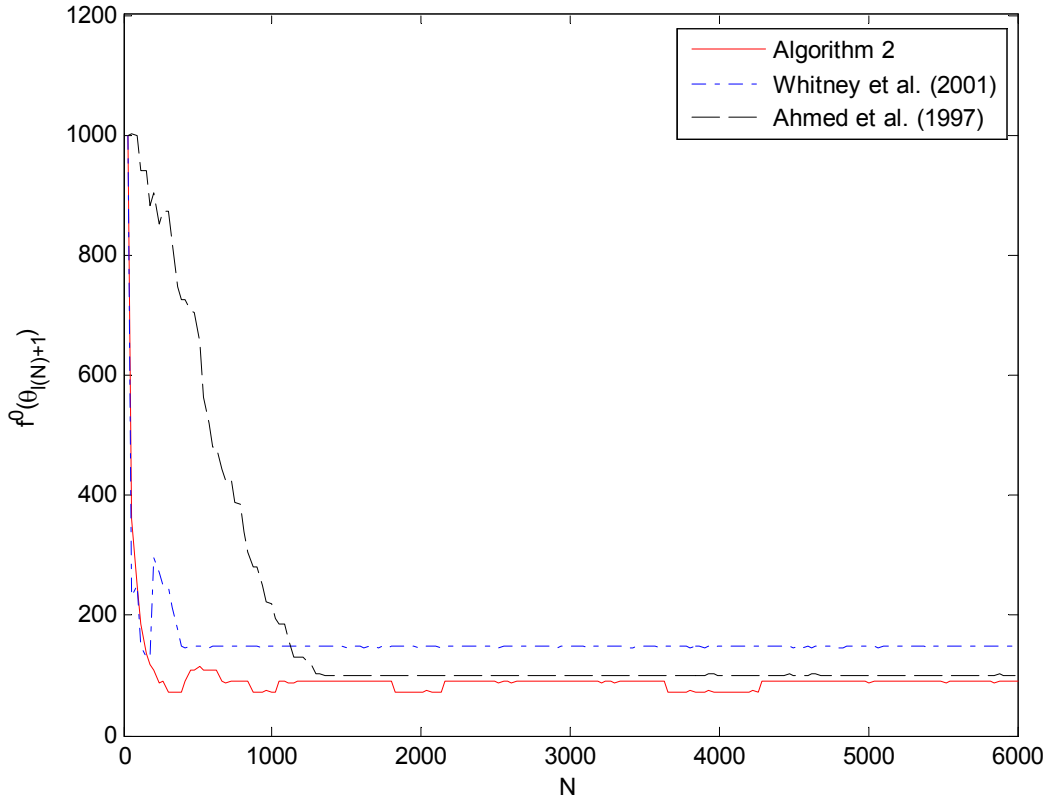

Figure 3.3: The plot of $f^0(\theta_{l(N)+1})$ versus N for the three methods.

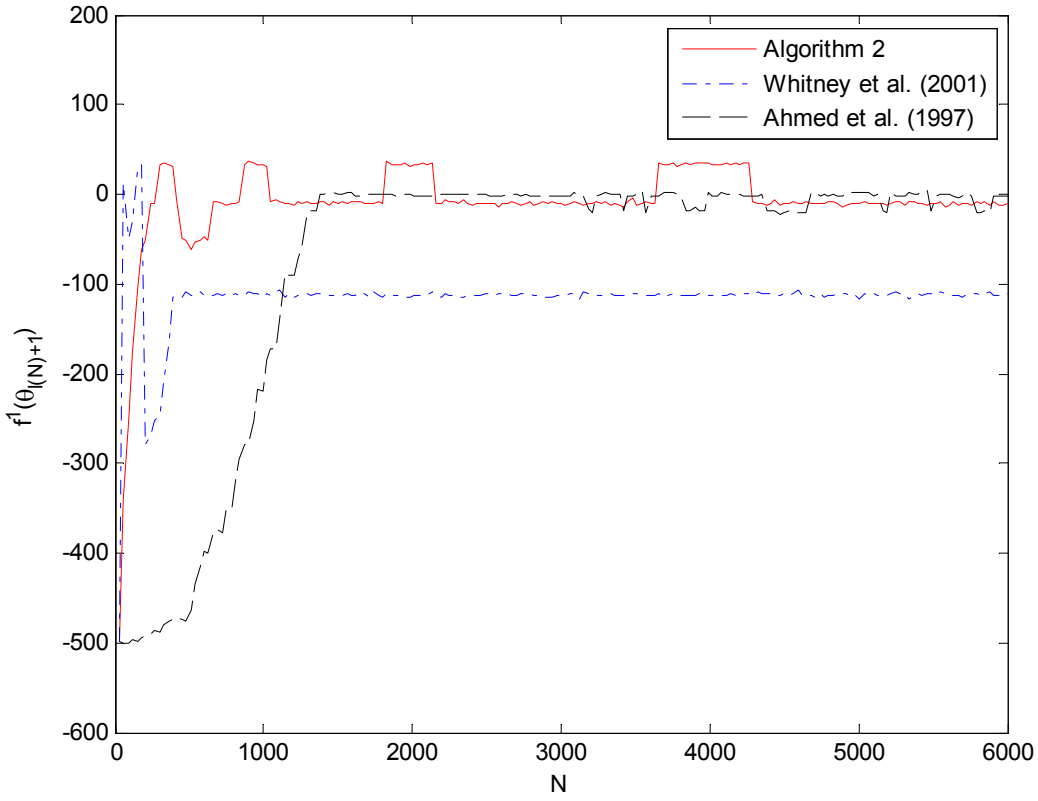

Figure 3.4: The plot of $f^1(\theta_{l(N)+1})$ versus N for the three methods.

3.4.3 Inventory Control in a Periodic Review System

We consider a finite-horizon, periodically-reviewed, single-item inventory system with integer-valued iid demands and full backlogging. Orders are received at the beginning of each period, the demand for the period arrives next, and we review the inventory position (= on hand stock minus backorders plus any outstanding orders) to make an ordering decision. The ordering decisions are made according to the (s, S) policy. If the inventory position is less than s, an order for the amount of S minus the inventory position is placed. Otherwise, no action is taken. The order lead time is assumed to be zero. When an order of x units is placed, the ordering cost of $K + cx$ is incurred, where K is the fixed setup cost per order and c is the unit cost. A holding cost of h per unit per period is charged against any unit left at the end of each time period. The

service level is measured using the fill rate, which is defined as the fraction of demand that is met directly from stock on hand. By $f^0(s, S)$, we denote the average ordering and holding costs per period over $1,000$ time periods when the inventory position at the beginning of the first period is initialized at S and the system is governed by the (s, S) policy. By $g(s, S)$, we denote the fill-rate over $1,000$ time periods when the inventory position at the beginning of the first period is initialized at S and the system is governed by the (s, S) policy. Our goal is to determine the values s and S, say s_* and S_*, that minimize $f^0(s, S)$ subject to the constraint that $g(s, S)$ is greater than or equal to a prescribed level β, i.e., $f^1(s, S) \triangleq \beta - g(s, S) \leq 0$.

We apply Algorithm 2, and the methods proposed by Whitney et al. (2001) and Ahmed et al. (1997) to find (s_*, S_*). Whenever we observe f^0 at each point in $\mathcal{F}_1 \triangleq \{(x, y) \in \mathbb{Z}^2 : 1 \leq x \leq 100, 1 \leq y \leq 100, x \leq y\}$ in all three methods, the inventory system is simulated over 1,000 time periods, the ordering and holding costs are averaged over the 1,000 time periods, and the average of 20 iid such replications is used as an observation of f^0. Likewise, whenever we observe g at each point in $\mathcal{F}_1$, the inventory system is simulated over 1,000 time periods, the demand which is met directly from stock over the 1,000 periods is divided by the total demand over the 1,000 periods, and the average of 20 iid such replications is used as an observation of g. θ_1 is set as $(100, 100)$ for all the three methods and $\lambda_1 = 275$ is used for Algorithm 2. The parameters used are $c_n = 500/(35 + n)$ for the first 10% of the total iterations available, $c_n = 50/(35 + n)$ for the rest of the iterations available, $a_n = 200/(35 + n), b_n = 10000 \log(n^{0.5}), \alpha = 0.95, T_f = 100, R = 0.6, M = 10$, $K = 100, c = 3, h = 3$, $\beta = 0.95$, and demand in each time period follows a Poisson distribution with a mean of 30.

To compare the estimates of (s_*, S_*) produced by the proposed method and other methods to the true values, s_* and S_* are estimated by evaluating $f^0(s, S)$ and $g(s, S)$, using the average of 100 iid replications at each point in $\mathcal{F}_1$ and selecting the values s

and S that minimize the estimated f^0 value while the estimated g value is greater than or equal to β. The "true" optimal solution estimated this way is $(s_*, S_*) = (18, 60)$.

Table 3.2: Averages (Mean) and standard deviation (Std) of $\theta_{l(N)+1}$ and corresponding averages of $f^1(\theta_{l(N)+1})$ generated from three methods applied to the periodically-reviewed inventory system.

	Algorithm 2			Whitney et al. (2001)			Ahmed et al. (1997)		
N	Mean	Std	$f^1(\theta)$ $(\times 10^{-3})$	Mean	Std	$f^1(\theta)$ $(\times 10^{-3})$	Mean	Std	$f^1(\theta)$ $(\times 10^{-3})$
1000	$(17, 62)$	5.2	9.0	$(42, 68)$	25.1	-16.8	$(88, 107)$	13.3	50.0
2000	$(17, 60)$	2.4	4.3	$(33, 75)$	21.0	-16.9	$(75, 106)$	19.9	49.6
4000	$(18, 60)$	0.7	0.8	$(28, 65)$	17.5	-9.2	$(34, 77)$	21.1	28.5
8000	$(18, 60)$	0.5	-0.1	$(18, 58)$	7.7	2.7	$(24, 66)$	17.0	10.3
12000	$(18, 60)$	0.4	-0.4	$(18, 58)$	6.3	6.8	$(21, 63)$	14.6	4.1
16000	$(18, 60)$	0.3	-0.7	$(17, 58)$	4.3	8.6	$(19, 61)$	13.5	2.4
20000	$(18, 60)$	0.3	-0.6	$(17, 58)$	4.0	8.0	$(18, 61)$	12.8	1.3
(s_*, S_*)	$(18, 60)$			$(18, 60)$			$(18, 60)$		

Denoting the number of simulation runs made at iteration n by t_n and fixing the total number N of simulation runs available, we compute $\theta_{l(N)+1}$ where $l(N)$ is the maximum number of iterations given the N simulation runs available; i.e., $l(N)$ is the largest integer satisfying $t_1 + \cdots + t_{l(N)} \leq N$. Thus, $\theta_{l(N)+1}$ is the best estimate of (s_*, S_*) given the computational budget N. We notice that both f^0 and g can be simultaneously computed in a single simulation run, so $t_n = 60, 60$, and 20 for Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997), respectively. Table 3.2 reports the averages (Mean) of $\theta_{l(N)+1}$ generated by Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997) based on 200 independent copies of $\theta_{l(N)}$ for each value of N. To measure how the distribution of $\theta_{l(N)+1}$ is spread out, the average of the sample standard deviation of $\theta^1_{l(N)+1}$ and $\theta^2_{l(N)+1}$ is reported in Table 3.2. In addition, averages of the $f^1(\theta_{l(N)+1})$

values are reported to show that our method converges to a feasible solution for N sufficiently large.

It is noteworthy that the extended functions of f^0 and g in the proposed method are not convex (see Song et al. (2008)), but the proposed method successfully finds the optimal solution nevertheless.

3.4.4 Staffing in a Call Center

We consider a call center which handles three types of calls; calls that request technical support, calls that ask for sales information, and calls that wish to check order status. An incoming call is one of the three types with probability 0.5, 0.3, and 0.2, respectively. Calls arrive at the call center according to a Poisson process with rate λ per minute. Calls that enter the center form a single queue of infinite capacity and are served on a first come first serve basis. The call center opens at 8 AM and closes at 6 PM. After 6 PM, all remaining calls should be handled before they exit the system. Thus, each simulation run starts and ends with an empty system.

If a customer requests technical support, they must select one of the three products (products 1, 2, and 3) that they wish technical support for. We assume that the percentages of requests for the three products are 25%, 34%, and 41%, respectively. The request for product i is served by a staff member of type i for $1 \leq i \leq 3$, and the service time per customer requested by a staff member of type i follows a triangular distribution with lower limit 3, upper limit 18, and mode 6. Staff members of type 4 are available to handle the calls for all three products. They serve a customer only when there are no staff members of types 1, 2, and 3 available. The service time per customer requested by a staff member of type 4 follows a triangular distribution with lower limit 3, upper limit 18, and mode 6.

If a customer asks for sales information, then the customer is serviced by a staff member of type 5. The service time per customer requested by a staff member of

type 5 follows a triangular distribution with lower limit 4, upper limit 45, and mode 15.

If a customer wishes to check order status, the request is handled by an automatic phone system, and there is no limit on the number of such calls that the automatic phone system can handle. The service time spent on the automated system follows a triangular distribution with lower limit 2, upper limit 4, and mode 3. After this automated service, 15% of the customers ask for a salesperson and wait on line until served by a staff member of type 5. The service time per customer requested by a staff member of type 5 in this case follows a triangular distribution with lower limit 4, upper limit 45, and mode 15.

Each staff member serves calls on a first come first serve basis. All service times are independent of each other and independent of the arrival process.

We denote the number of staff members of type i by θ^i $(1 \leq i \leq 5)$. By $f^0(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$, we denote the daily average operating costs, given the staffing level $(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$. By $g(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$, we denote the fraction of calls waiting less than 90 seconds in the queue before they initiate their service, given the staffing level $(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$. The goal is to find the numbers of staff members of types $1, 2, 3, 4,$ and 5 minimizing $f^0(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$ while ensuring that $f^1(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5) \triangleq 0.8 - g(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5) \leq 0$. The operating costs consist of labor costs, which are \$100 per day for staff members of types 1, 2, 3, and 5, and \$200 per day for a staff member of type 4.

We apply Algorithm 2, and the methods proposed by Whitney et al. (2001) and Ahmed et al. (1997) to find the optimal values of θ^i, say θ^i_* $(1 \leq i \leq 5)$. Whenever we observe f^0 at each point in $\mathcal{F}_2 \triangleq \{\theta \in \mathbb{Z}^5 : 1 \leq \theta^i \leq 100 \text{ for } 1 \leq i \leq 5\}$ in all three methods, we simulate the system over 10 days, and compute the average of the operating costs over the 10–day time horizon, and use the average of 10 iid such replications as an observation of f^0. Likewise, whenever we observe the fraction

of calls that wait less than 90 seconds in the queue, we simulate the system over 10 days, divide the number of calls that waited less than 90 seconds in the queue by the total number of calls over the 10–day time horizon, and use the average of 10 iid such replications as an observation of g. θ_1 is set as $(50, 50, 50, 50, 50)$ for all the three methods and $\lambda_1 = 0$ is used for Algorithm 2. The parameters used are $c_n = 0.5/(33 + n)$, $a_n = 2/(50 + n), b_n = 200000 \log(n^{0.5}), \alpha = 0.95, T_f = 50, R = 0.6, M = 15$, and $\lambda = 5$.

The optimal policy $(\theta^1_*, \theta^2_*, \theta^3_*, \theta^4_*, \theta^5_*)$ is estimated by evaluating $f^0(\theta)$ and the fraction of calls waited less than 90 seconds, using the average of 50 iid observations at each $\theta \in \mathcal{F}_2$. The "true" optimal solution estimated this way is $(5, 15, 19, 0, 31)$.

Denoting the number of simulation runs made at iteration n by t_n and fixing the total number N of simulation runs available, we compute $\theta_{l(N)+1}$ where $l(N)$ is the maximum number of iterations given the N simulation runs available; i.e., $l(N)$ is the largest integer satisfying $t_1 + \cdots + t_{l(N)} \leq N$. Thus, $\theta_{l(N)+1}$ is the best estimate of $(\theta^1_*, \theta^2_*, \theta^3_*, \theta^4_*, \theta^5_*)$ given the computational budget N. We note that both f^0 and g can be simultaneously computed in a single simulation run, so $t_n = 60, 30$, and 10 for Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997), respectively. Table 3.3 reports the averages (Mean) of $\theta_{l(N)+1}$ generated by Algorithm 2, the method by Whitney et al. (2001), and the method by Ahmed et al. (1997) based on 50 independent copies of $\theta_{l(N)}$ for each value of N. To measure how the distribution of $\theta_{l(N)+1}$ is spread out, the average of the sample standard deviation of $\theta^1_{l(N)+1}, \theta^2_{l(N)+1}, \theta^3_{l(N)+1}, \theta^4_{l(N)+1}$, and $\theta^5_{l(N)+1}$ is reported in Table 3.3. In addition, averages of the $f^1(\theta_{l(N)+1})$ values are reported to show that our method converges to a feasible solution for N sufficiently large.

Table 3.3: Averages (Mean) and standard deviation (Std) of $\theta_{l(N)+1}$ and corresponding averages of $f^1(\theta_{l(N)+1})$ generated from three methods applied to the call center.

	Algorithm 2		
N	Mean	Std	$f^1(\theta)$
1000	(30, 30, 30, 10, 30)	0.00	-0.10
2000	(15, 16, 19, 0, 31)	2.12	-0.03
3000	(10, 15, 19, 4, 35)	5.50	-0.04
4000	(8, 15, 18, 3, 34)	5.43	-0.02
5000	(8, 15, 18, 3, 33)	4.83	-0.01
6000	(8, 14, 17, 2, 33)	4.60	0.00
7000	(8, 14, 17, 2, 33)	4.53	-0.02
8000	(8, 14, 17, 1, 33)	4.03	-0.01
10000	(8, 14, 17, 1, 32)	3.22	-0.01
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(5, 15, 19, 0, 31)		

	Whitney et al. (2001)		
N	Mean	Std	$f^1(\theta)$
1000	(19, 27, 28, 12, 41)	23.47	0.01
2000	(19, 24, 28, 8, 39)	19.10	0.01
3000	(19, 21, 23, 9, 40)	16.76	-0.01
4000	(14, 22, 17, 9, 38)	15.67	-0.01
5000	(17, 21, 21, 7, 37)	12.86	-0.03
6000	(16, 18, 22, 7, 37)	14.96	-0.01
7000	(16, 16, 21, 7, 39)	12.48	-0.02
8000	(15, 14, 19, 6, 36)	10.62	0.00
10000	(16, 14, 19, 6, 38)	9.45	-0.02
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(5, 15, 19, 0, 31)		

	Ahmed et al. (1997)		
N	Mean	Std	$f^1(\theta)$
1000	(47, 44, 45, 44, 47)	2.15	-0.17
2000	(43, 38, 38, 38, 44)	3.01	-0.15
3000	(40, 31, 32, 32, 41)	3.53	-0.14
4000	(37, 25, 26, 27, 38)	3.94	-0.13
5000	(33, 18, 19, 21, 35)	4.12	-0.12
6000	(31, 12, 13, 14, 31)	4.44	-0.08
7000	(28, 9, 10, 11, 30)	4.28	0.01
8000	(27, 8, 9, 10, 30)	4.32	0.03
10000	(26, 8, 9, 10, 30)	4.31	0.03
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(5, 15, 19, 0, 31)		

3.4.5 Staffing in an Emergency Room

We consider an emergency department in a hospital which operates 24 hours a day and receives two kinds of patients: walk-in patients who are required to see a receptionist before entering the queue to the examination room, and patients who are delivered by ambulances and so can enter the queue to the examination room directly without seeing a receptionist.

The arrival process of the walk-in patients forms a non-homogeneous Poisson process with the rate parameter $\lambda(t)$ given as follows:

$$\lambda(t) = \begin{cases} 5, & \text{if } 0 \le t < 2 \\ 4, & \text{if } 2 \le t < 4 \\ 3, & \text{if } 4 \le t < 6 \\ 5, & \text{if } 6 \le t < 8 \\ 7, & \text{if } 8 \le t < 10 \\ 8, & \text{if } 10 \le t < 12 \\ 9, & \text{if } 12 \le t < 14 \\ 8, & \text{if } 14 \le t < 20 \\ 6, & \text{if } 20 \le t < 22 \\ 3, & \text{if } 22 \le t < 24 \end{cases} \tag{3.24}$$

per hour. The arrival process of the patients delivered by ambulances forms a homogeneous Poisson process with a rate of 2 per hour.

At the examination room, one of the doctors examines the patient and decides whether any tests are necessary to give a diagnosis; if so, the patient enters the queue to a lab, where one of the lab technicians performs necessary tests. Once the patient is released from the lab, he or she re-enters the queue to the examination room to get the test results from a doctor. The test results are ready immediately after the tests are conducted, but the patient must wait in the queue to the examination room

to get the doctor's opinion on the test results. Based on the test results, the doctor decides on one of the three types of treatments for the patient: (1) a patient can take normal treatment, which is performed by the nurses in the treatment room, (2) a patient can take emergency treatment, which is performed by the nurses in the emergency room, and (3) a patient can be released from the hospital after receiving their medication. If a patient does not need any tests, then the doctor decides on one of the three types of treatments for the patient the first time the patient visits the doctor in the examination room. Thus, a patient receives an opinion from a doctor after getting tests in the lab or at the first visit to the examination room. In both cases, a patient receives a treatment of types (1), (2), and (3) with probabilities 0.4, 0.4, and 0.2, respectively.

All queues are assumed to have infinite capacity and the group of receptionists (or groups of doctors in the examination room, lab technicians at the lab, nurses in the treatment room, and nurses in the emergency room, respectively) forms multi-servers serving a single common queue of patients.

All services are based on a first come first serve basis and all service times are independent of each other and independent of the arrival processes.

The service times at the receptionists' desk follow an exponential distribution with a mean of 7.5. The service times at the examination room follow an exponential distribution with a mean of 15. The service times at the lab follow a triangular distribution with lower limit 10, upper limit 30, and mode 20. The service times at the treatment room follow a triangular distribution with lower limit 20, upper limit 30, and mode 28. The service times at the emergency room follow an exponential distribution with a mean of 90.

The goal is to find the numbers of receptionists, doctors, lab technicians, nurses in the treatment room, and nurses in the emergency room, denoted by θ^1, θ^2, θ^3, θ^4, and θ^5, respectively, that minimize the average operating cost $f^0(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$ per day

over a 150–day period while ensuring that the probability of a patient who receives a treatment of type 2 waiting less than 1 hour in the queue, $g(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$, is greater than or equal to 0.9, i.e., $f^1(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5) \triangleq 0.9 - g(\theta^1, \theta^2, \theta^3, \theta^4, \theta^5) \leq 0$. The daily operational costs consist of labor costs which are \$150 per receptionist per day, \$1200 per doctor per day, \$500 per lab technician per day, \$350 per nurse in the treatment room per day, and \$350 per nurse in the emergency room per day.

We apply Algorithm 2, and the methods proposed by Whitney et al. (2001) and Ahmed et al. (1997) to find the optimal values of θ^i, say θ^i_* $(1 \leq i \leq 5)$. Whenever we observe f^0 at each point in $\mathcal{F}_3 \triangleq \{\theta \in \mathbb{Z}^5 : 1 \leq \theta^i \leq 50, 1 \leq i \leq 5\}$ in all three methods, the system is simulated over 150 time periods, the operating costs are averaged over the 150 time periods, and the average of 10 iid such replications is used as an observation of f^0. Likewise, whenever we observe g, the system is simulated over 150 time periods, the number of patients who receive treatment of type 2 and spend less than 1 hour in the queue is divided by the total number patients of type 2 over the 150 periods, and the average of 10 iid such replications is used as an observation of g. θ_1 is set as $(30, 30, 30, 30, 30)$ for all the three methods and $\lambda_1 = 0$ is used for Algorithm 2. The parameters used are $c_n = 0.3/(100 + n)$ for the first 50% of the total iterations available, $c_n = 0.1/(100 + n)$ for the rest of the iterations available, $a_n = 0.3/(50 + n), b_n = 5000 \log(n^{0.5}), \alpha = 0.95, T_f = 50, R = 0.6$, and $M = 5$.

The optimal policy $(\theta^1_*, \theta^2_*, \theta^3_*, \theta^4_*, \theta^5_*)$ is estimated by evaluating $f^0(\theta)$ and $g(\theta)$, using the average of 50 iid observations at each $\theta \in \mathcal{F}_3$. The "true" optimal solution estimated this way is $(2, 5, 2, 1, 10)$.

Table 3.4: Averages (Mean) and standard deviation (Std) of $\theta_{l(N)+1}$ and corresponding averages of $f^1(\theta_{l(N)+1})$ generated from three methods applied to the emergency department.

	Algorithm 2		
N	Mean	Std	$f^1(\theta)$
1000	(23, 5, 8, 14, 14)	0.00	-0.05
2000	(17, 5, 3, 1, 8)	0.49	0.18
3000	(12, 5, 4, 1, 8)	0.46	0.02
4000	(7, 5, 4, 1, 8)	0.54	0.01
5000	(3, 5, 4, 1, 8)	0.52	0.01
6000	(3, 5, 3, 1, 8)	0.20	0.02
7000	(2, 5, 2, 1, 8)	0.20	0.03
8000	(2, 5, 2, 1, 8)	0.20	0.03
10000	(2, 5, 3, 1, 8)	0.19	0.02
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(2, 5, 2, 1, 10)		

	Whitney et al. (2001)		
N	Mean	Std	$f^1(\theta)$
1000	(18, 5, 4, 8, 14)	7.36	0.27
2000	(16, 6, 4, 5, 11)	5.60	0.20
3000	(15, 5, 4, 5, 10)	4.41	0.19
4000	(10, 5, 3, 4, 9)	3.81	0.19
5000	(8, 5, 3, 3, 9)	3.26	0.22
6000	(7, 5, 2, 3, 9)	2.93	0.21
7000	(7, 5, 2, 2, 9)	2.57	0.26
8000	(6, 5, 2, 2, 8)	2.03	0.20
10000	(5, 5, 2, 2, 8)	1.75	0.18
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(2, 5, 2, 1, 10)		

	Ahmed et al. (1997)		
N	Mean	Std	$f^1(\theta)$
1000	(28, 27, 27, 27, 29)	1.35	-0.10
2000	(27, 24, 23, 24, 27)	2.02	-0.10
3000	(26, 20, 20, 21, 26)	2.44	-0.10
4000	(24, 18, 17, 18, 24)	2.80	-0.10
5000	(23, 15, 14, 15, 23)	3.08	-0.10
6000	(22, 12, 11, 12, 21)	3.42	-0.09
7000	(20, 10, 8, 8, 19)	3.57	-0.08
8000	(19, 8, 5, 5, 17)	3.48	-0.06
10000	(15, 6, 3, 2, 14)	2.58	-0.03
$(\theta_*^1,\theta_*^2,\theta_*^3,\theta_*^4,\theta_*^5)$	(2, 5, 2, 1, 10)		

Denoting the number of simulation runs made at iteration n by t_n and fixing the total number N of simulation runs available, we compute $\theta_{l(N)+1}$ where $l(N)$ is the maximum number of iterations given the N simulation runs available; i.e., $l(N)$ is the largest integer satisfying $t_1 + \cdots + t_{l(N)} \leq N$. Thus, $\theta_{l(N)+1}$ is the best estimate of θ_* given the computational budget N. We note that both f^0 and g can be simultaneously computed in a single simulation run, so $t_n = 60, 30$, and 10 for Algorithm 2, the method proposed by Whitney et al. (2001), and the method proposed by Ahmed et al. (1997), respectively. Table 3.4 reports the averages (Mean) of $\theta_{l(N)+1}$ generated by Algorithm 2, the method proposed by Whitney et al. (2001), and the method proposed by Ahmed et al. (1997) based on 50 independent copies of $\theta_{l(N)}$ for each value of N. To measure how the distribution of $\theta_{l(N)+1}$ is spread out, the average of the sample standard deviation of $\theta^1_{l(N)+1}, \theta^2_{l(N)+1}, \theta^3_{l(N)+1}, \theta^4_{l(N)+1}$, and $\theta^5_{l(N)+1}$ is reported in Table 3.4. In addition, averages of the $f^1(\theta_{l(N)+1})$ values are reported to show that our method converges to a feasible solution for N sufficiently large.

In all three examples, our methods display good performance.

3.5. Proof of Theorem 1

Our proof of Theorem 1 can be broken down into a number of key steps. In Step 1, we prove i) and ii) of Theorem 1. To prove iii) of Theorem 1, we first show that both $\widehat{f^i}$ and $\widehat{L}$ are uniformly bounded in Step 2. We then show the expected value of a subgradient with respect to θ is bounded in Step 3. With these two conditions and Lemma 2 in p. 344 of Benveniste et al. (1990), we are able to show the sequence

$$Z(\theta_n, \lambda_n) = \|\theta_n - \widehat{\theta}_*\|^2 + \|\lambda_n - \widehat{\lambda}_*\|^2$$

is uniformly bounded and converges in Steps 4 and 5. Finally, we follow the proof of the theorem in p. 378 of Kushner and Sanvicente (1975) to show $\theta_n \to \widehat{\theta}_*$ a.s. as $n \to \infty$ in Steps 6, 7 and 8.

Step 1 By A2 and A3, there exists a saddle point $(\widehat{\theta}_*, \widehat{\lambda}_*)$ of $\widehat{L}$, i.e.,

$$\widehat{L}(\widehat{\theta}_*, \lambda) \leq \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \leq \widehat{L}(\theta, \widehat{\lambda}_*)$$

for $\theta \in C$ and $\lambda \in \mathbb{R}^r_+$ (see Theorem 1 in p. 217 of Luenberger (1969)). The uniqueness of $\widehat{\theta}_*$ follows from the strict convexity of $\widehat{f}^0$. (Suppose, on the contrary, there exists a saddle point $(\overline{\theta}, \overline{\lambda})$ of $\widehat{L}$ such that $\overline{\theta} \neq \widehat{\theta}_*$, then $\widehat{f}^0(\widehat{\theta}_*) = \widehat{f}^0(\overline{\theta})$, which contradicts the strict convexity of $\widehat{f}^0$.) In fact, if $\widehat{\theta}_*$ is an integer point, then by Proposition 1, $\widehat{\theta}_*$ is an optimal solution to (3.1). Furthermore, (3.1) has a unique solution. To see why the optimal solution to (3.1) is unique, suppose that there exists an optimal solution $\theta' \in \mathbb{Z}^d$ to (3.1) such that $\theta' \neq \widehat{\theta}_*$. For any $0 < t < 1$, define $\theta_t = t\theta' + (1-t)\widehat{\theta}_*$. Then θ_t is a feasible solution to (3.8). By the strict convexity of $\widehat{f}^0$, we have

$$\begin{aligned} \widehat{f}^0(\theta_t) &< t\widehat{f}^0(\theta') + (1-t)\widehat{f}^0(\widehat{\theta}_*) \\ &= tf^0(\theta') + (1-t)f^0(\widehat{\theta}_*) \\ &= f^0(\widehat{\theta}_*) \\ &= \widehat{f}^0(\widehat{\theta}_*), \end{aligned}$$

which contradicts the fact that $\widehat{\theta}_*$ is an optimal solution to (3.8).

Step 2 We observe that $\widehat{f}^i$ is uniformly bounded on $\mathcal{B}_\theta$ for $0 \leq i \leq r$ and $\widehat{L}$ is uniformly bounded on $\mathcal{B}$. To see why this is true, we note that $\widehat{f}^i$ is convex on $\mathbb{R}^d$ and hence is continuous on $\mathcal{B}_\theta$ (see Theorem 10.1 in p. 82 of Rockafellar (1970)). By the compactness of $\mathcal{B}_\theta$, $\widehat{f}^i$ is uniformly bounded on $\mathcal{B}_\theta$ (see Theorem 4.4.1 in p.

189 of Marsden and Hoffman (1993)). By the compactness of $\mathcal{B}$, $\widehat{L}$ is also uniformly bounded. In fact, $\widehat{L}$ is bounded on any compact subset of $\mathbb{R}^d \times \mathbb{R}^r$.

Step 3 We let $d_n = \mathbb{E}\left[D_n(\theta_n, \lambda_n)|\mathcal{F}_n\right]$ and observe that $d_n(\theta_n, \lambda_n)$ is bounded on $\mathcal{B}$. To see why this is true, let ξ_n be the d–dimensional vector whose jth component is 1 if the jth component of d_n is nonnegative and -1 otherwise. Thus $d_n^T \xi_n = \sum_{j=1}^d |d_n^j|$, where d_n^j is the jth component of d_n. From (3.10), we obtain

$$\widehat{L}(\theta_n + \xi_n, \lambda_n) - \widehat{L}(\theta_n, \lambda_n) \geq d_n^T \xi_n = \sum_{j=1}^{d} |d_n^j|.$$

Since $\widehat{L}$ is bounded on any compact subset of $\mathbb{R}^d \times \mathbb{R}^r$ and $(\theta_n, \lambda_n) \in \mathcal{B}$, $\widehat{L}(\theta_n + \xi_n, \lambda_n) - \widehat{L}(\theta_n, \lambda_n)$ is bounded. Thus $\sum_{j=1}^d |d_n^j|$ and $\|d_n\|$ are bounded.

Step 4 We let $Z : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}$ be defined by

$$Z(\theta, \lambda) = \|\theta - \widehat{\theta}_*\|^2 + \|\lambda - \widehat{\lambda}_*\|^2$$

for $(\theta, \lambda) \in \mathbb{R}^d \times \mathbb{R}^r$.

We prove that

$$\mathbb{E}\left[Z(\theta_{n+1}, \lambda_{n+1})|\mathcal{F}_{n+1}\right] - Z(\theta_n, \lambda_n) \quad \leq \quad -2c_n Q(\theta_n, \lambda_n) + Cc_n^2 \tag{3.25}$$

for $n \geq 1$ and for some positive constant C, where

$$Q(\theta_n, \lambda_n) = (\theta_n - \widehat{\theta}_*)^T d_n - (\lambda_n - \widehat{\lambda}_*)^T (\widetilde{f}^i(\theta_n) : 1 \leq i \leq r)$$

and that

$$Q(\theta_n, \lambda_n) \quad > \quad 0 \tag{3.26}$$

for $\theta_n \neq \widehat{\theta}_*$.

To prove (3.25), we first note

$$\|\theta_{n+1} - \widehat{\theta}_*\|^2 \;\leq\; \|\overline{\theta}_{n+1} - \widehat{\theta}_*\|^2 \tag{3.27}$$

because θ_{n+1} is the projection of $\overline{\theta}_{n+1}$ onto $\mathcal{B}_\theta \triangleq \left\{\theta \in \mathbb{R}^d : |\theta^i| \leq K \text{ for } 1 \leq i \leq d\right\}$ and $\widehat{\theta}_*$ is in $\mathcal{B}_\theta$, θ_{n+1} is no further from $\widehat{\theta}_*$ than is $\overline{\theta}_{n+1}$. By a similar reasoning, we have

$$\|\lambda_{n+1} - \widehat{\lambda}_*\|^2 \;\leq\; \|\overline{\lambda}_{n+1} - \widehat{\lambda}_*\|^2. \tag{3.28}$$

From (3.27),

$$\begin{aligned}
&\mathbb{E}\left[\|\theta_{n+1} - \widehat{\theta}_*\|^2|\mathcal{F}_n\right] - \|\theta_n - \widehat{\theta}_*\|^2 \\
&\leq\; \mathbb{E}\left[\|\overline{\theta}_{n+1} - \widehat{\theta}_*\|^2|\mathcal{F}_n\right] - \|\theta_n - \widehat{\theta}_*\|^2 \\
&=\; \mathbb{E}\left[\|\theta_n - c_n D_n(\theta_n, \lambda_n) - \widehat{\theta}_*\|^2|\mathcal{F}_n\right] - \|\theta_n - \widehat{\theta}_*\|^2 \\
&=\; -2c_n(\theta_n - \widehat{\theta}_*)^T d_n + c_n^2 \mathbb{E}\left[\|D_n(\theta_n, \lambda_n)\|^2|\mathcal{F}_n\right] \\
&\leq\; -2c_n(\theta_n - \widehat{\theta}_*)^T d_n + 4c_n^2\|d_n\|^2 + 4c_n^2\sigma^2 \quad \text{by (3.12)} \\
&\leq\; -2c_n(\theta_n - \widehat{\theta}_*)^T d_n + C_1 c_n^2 \quad \text{by Step 3}
\end{aligned} \tag{3.29}$$

for some positive constant C_1. The second last inequality follows because $\|x_1 + \cdots + x_l\|^m \leq l^m(\|x_1\|^m + \cdots + \|x_l\|^m)$ for $x_1, \ldots, x_l \in \mathbb{R}^d$ and positive integers l and m.

On the other hand, from (3.28) we obtain

$$\begin{aligned}
&\mathbb{E}\left[\|\lambda_{n+1} - \widehat{\lambda}_*\|^2|\mathcal{F}_n\right] - \|\lambda_n - \widehat{\lambda}_*\|^2 \\
&\leq\; \mathbb{E}\left[\|\overline{\lambda}_{n+1} - \widehat{\lambda}_*\|^2|\mathcal{F}_n\right] - \|\lambda_n - \widehat{\lambda}_*\|^2 \\
&\leq\; 2c_n(\lambda_n - \widehat{\lambda}_*)^T(\widehat{f}^i(\theta_n) : 1 \leq i \leq r) + 2c_n^2 \sum_{i=1}^{r}(\widehat{f}^i(\theta_n))^2 + 2c_n^2 r\sigma^2 \\
&\leq\; 2c_n(\lambda_n - \widehat{\lambda}_*)^T(\widehat{f}^i(\theta_n) : 1 \leq i \leq r) + C_2 c_n^2 \quad \text{by Step 2}
\end{aligned} \tag{3.30}$$

for some positive constant C_2.

Equations (3.29) and (3.30) combine to yield (3.25).

Next we prove (3.26). By (3.10),

$$\widehat{L}(\widehat{\theta}_*, \lambda_n) \;\geq\; \widehat{L}(\theta_n, \lambda_n) + (\widehat{\theta}_* - \theta_n)^T d_n. \tag{3.31}$$

On the other hand, by the definition of the Lagrangian,

$$\widehat{L}(\theta_n, \widehat{\lambda}_*) - (\widehat{\lambda}_* - \lambda_n)^T (\widehat{f}^i(\theta) : 1 \leq i \leq r) \;=\; \widehat{L}(\theta_n, \lambda_n). \tag{3.32}$$

From (3.31) and (3.32), we get

$$\begin{aligned} \widehat{L}(\widehat{\theta}_*, \lambda_n) - \widehat{L}(\theta_n, \widehat{\lambda}_*) &\geq (\widehat{\theta}_* - \theta_n)^T d_n - (\widehat{\lambda}_* - \lambda_n)^T (\widehat{f}^i(\theta) : 1 \leq i \leq r) \\ &= -Q(\theta_n, \lambda_n). \end{aligned} \tag{3.33}$$

By the definition of the saddle point, we have

$$\widehat{L}(\widehat{\theta}_*, \lambda_n) \;\leq\; \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \leq \widehat{L}(\theta_n, \widehat{\lambda}_*). \tag{3.34}$$

By the strict convexity of $\widehat{f}^0$, for $\theta_n \neq \widehat{\theta}_*$, we obtain

$$\widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) \;<\; \widehat{L}(\theta_n, \widehat{\lambda}_*) \tag{3.35}$$

because otherwise $\widehat{L}(\eta, \widehat{\lambda}_*)$, as a function of η, will be constant on the line segment connecting θ_n and $\widehat{\theta}_*$, contradicting the strict convexity of $\widehat{f}^0$. Combining (3.34) and (3.35) yields

$$0 \;>\; \widehat{L}(\widehat{\theta}_*, \lambda_n) - \widehat{L}(\theta_n, \widehat{\lambda}_*). \tag{3.36}$$

Hence, (3.26) follows from (3.33) and (3.36).

Step 5 We observe that applying Lemma 2 in p. 344 of Benveniste et al. (1990) to the sequence $(Z(\theta_n, \lambda_n) : n \geq 1)$ yields

$$\sum_{n=1}^{\infty} c_n Q(\theta_n, \lambda_n) \quad < \quad \infty \tag{3.37}$$

a.s. as $n \to \infty$ and $Z(\theta_n, \lambda_n) \to Z_\infty$ a.s. for some finite-valued random variable Z_∞ as $n \to \infty$.

Step 6 We prove that for any $\epsilon > 0$, there exists $\delta > 0$ such that $Q(\theta_n, \lambda_n) \geq \delta$ whenever $\|\theta_n - \widehat{\theta}_*\| \geq \epsilon$.

To fill in the details, let

$$\delta = \inf\{\widehat{L}(\theta, \widehat{\lambda}_*) - \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*) : \theta \in \mathcal{B}_\theta, \|\theta - \widehat{\theta}_*\| \geq \epsilon\}$$

for any given $\epsilon > 0$. If $\delta = 0$, then there exists a sequence $(\theta_s : s \geq 1)$ in $\mathcal{B}_\theta$ with $\widehat{L}(\theta_s, \widehat{\lambda}_*) \to \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*)$ as $s \to \infty$. Since the θ_ss are bounded, there exists a subsequence $(\theta_{s_k} : k \geq 1)$ converging to a point θ_0 in $\{\theta \in \mathcal{B}_\theta : \|\theta - \widehat{\theta}_*\| \geq \epsilon\}$ such that $\widehat{L}(\theta_{s_k}, \widehat{\lambda}_*) \to \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*)$ as $k \to \infty$. By the continuity of $\widehat{L}$, $\widehat{L}(\theta_{s_k}, \widehat{\lambda}_*) \to \widehat{L}(\theta_0, \widehat{\lambda}_*)$ as $k \to \infty$ and hence $\widehat{L}(\theta_0, \widehat{\lambda}_*) = \widehat{L}(\widehat{\theta}_*, \widehat{\lambda}_*)$, but $\theta_0 \neq \widehat{\theta}_*$. This contradicts the uniqueness of $\widehat{\theta}_*$.

The rest of the proof is similar to the proof of the Theorem in p. 378 of Kushner and Sanvicente (1975). However, to make this proof self-contained, we present a complete argument.

Step 7 We show that for any $\epsilon > 0$, $\|\theta_n - \widehat{\theta}_*\| \leq 3\epsilon$ for all but finitely many n a.s.

Let $\epsilon > 0$ be given. By Step 6, we have $\|\theta_n - \widehat{\theta}_*\| \leq \epsilon$ for infinitely many n a.s. because otherwise, $\sum_{n=1}^{\infty} c_n Q(\theta_n, \lambda_n) \to \infty$ for some non-null set, which contradicts (3.37). First we show that $|c_n(D_n(\theta_n, \lambda_n) - d_n)| \geq \epsilon/2$ for finitely many n a.s. To see

this, note that

$$\begin{aligned}
&\mathbb{P}(\|c_n(D_n(\theta_n,\lambda_n)-d_n)\|\geq\epsilon/2)\\
&\quad\leq\quad (4c_n^2/\epsilon^2)\mathbb{E}\left[\|D_n(\theta_n,\lambda_n)-d_n\|^2\right]\quad\text{by Markov inequality}\\
&\quad=\quad (4c_n^2/\epsilon^2)\mathbb{E}\left[\mathbb{E}\left[\|D_n(\theta_n,\lambda_n)-d_n\|^2|\mathcal{F}_n\right]\right]
\end{aligned}$$

and that

$$\mathbb{E}\left[\|D_n(\theta_n,\lambda_n)-d_n\|^2|\mathcal{F}_n\right]\leq\sigma^2$$

by (3.12).

So it follows

$$\mathbb{P}(\|c_n(D_n(\theta_n,\lambda_n)-d_n)\|\geq\epsilon/2)\quad\leq\quad 4\sigma^2c_n^2/\epsilon^2.$$

Because $\sum_{n=1}^{\infty}c_n^2<\infty$, the Borel–Cantelli lemma guarantees $\|c_n(D_n(\theta_n,\lambda_n)-d_n)\|\geq\epsilon/2$ for finitely many n a.s.

We consider n sufficiently large so that $c_n\|d_n\|<\epsilon/2$, then we get $\|\theta_{n+1}-\theta_n\|\leq\|\overline{\theta}_{n+1}-\theta_n\|=c_n\|D_n(\theta_n,\lambda_n)-d_n+d_n\|\leq\epsilon$ for all but finitely many n.

We define the sets

$$\begin{aligned}
\mathcal{N}_\epsilon &= \{\theta\in\mathbb{R}^d:\|\theta-\widehat{\theta}_*\|\leq\epsilon\},\\
\mathcal{C}_{3\epsilon} &= \{\theta\in\mathbb{R}^d:2\epsilon\leq\|\theta-\widehat{\theta}_*\|\leq3\epsilon\},\\
\mathcal{N}_{3\epsilon} &= \{\theta\in\mathbb{R}^d:\|\theta-\widehat{\theta}_*\|\leq3\epsilon\},\\
\mathcal{N}_{4\epsilon} &= \{\theta\in\mathbb{R}^d:\|\theta-\widehat{\theta}_*\|\leq4\epsilon\}.
\end{aligned}$$

We let ϵ be small enough so that $\mathcal{N}_{4\epsilon}$ is in $\mathcal{B}_\theta$. Note that each time θ_n goes from $\mathcal{N}_\epsilon$ to the exterior of $\mathcal{N}_{3\epsilon}$, it must enter $\mathcal{C}_{3\epsilon}$ before ever going to the exterior of $\mathcal{N}_{3\epsilon}$ because θ_n cannot take a step larger than ϵ.

We define

$$\begin{aligned} t_1 &= \min\{n \geq 1 : \theta_n \in \mathcal{C}_{3\epsilon}\} \\ t_1^+ &= \min\{n \geq t_1 : \theta_n \in \mathcal{N}_\epsilon \text{ or exterior of } \mathcal{N}_{4\epsilon}\} \end{aligned}$$

and, inductively,

$$\begin{aligned} t_m &= \min\{n > t_{m-1}^+ : \theta_n \in \mathcal{C}_{3\epsilon}\} \\ t_m^+ &= \min\{n > t_m : \theta_n \in \mathcal{N}_\epsilon \text{ or exterior of } \mathcal{N}_{4\epsilon}\} \end{aligned}$$

for $m \geq 2$. The t_ms and t_m^+s are set equal to ∞ if not otherwise defined. We note that if t_m is finite, then t_m^+ is also finite a.s. because θ_n visits $\mathcal{N}_\epsilon$ infinitely many times a.s. By Step 4, for any positive integer m and $t_m < \infty$, $Q(\theta_n, \lambda_n) \geq \delta$ for $n = t_m, \ldots, t_m^+ - 1$.

We will show that t_m is finite for finitely many m a.s. Then it will follow that $\mathcal{C}_{3\epsilon}$ is entered for finitely many n a.s. and hence θ_n can leave $\mathcal{N}_{3\epsilon}$ finitely many times a.s., proving $\theta_n \to \widehat{\theta}_*$ a.s. as $n \to \infty$.

We let $I_{\{t_m<\infty\}}$ be 1 if t_m is finite and 0 otherwise. It will be shown in Step 8 that

$$\liminf_{m\to\infty} I_{\{t_m<\infty\}} \sum_{n=t_m}^{t_m^+-1} c_n \quad \geq \quad \alpha \liminf_{m\to\infty} I_{\{t_m<\infty\}} \tag{3.38}$$

for some positive constant α.

Equation (3.38) implies that if $t_m < \infty$ infinitely often, then it follows

$$\sum_{n=1}^{\infty} c_n Q(\theta_n, \lambda_n) \geq \sum_{m=1}^{\infty} I_{\{t_m<\infty\}} \sum_{n=t_m}^{t_m^+-1} c_n Q(\theta_n, \lambda_n) \geq \delta \sum_{m=1}^{\infty} I_{\{t_m<\infty\}} \sum_{n=t_m}^{t_m^+-1} c_n = \infty$$

by Step 6. So, $t_m < \infty$ infinitely often on some null set.

Step 8 We now prove (3.38).

We let

$$W_{s,t} = \sum_{i=s}^{t} c_i \left(D_i(\theta_i, \lambda_i) - d_i \right)$$

and note that $(W_{s,t} : t \geq s)$ is a martingale for each fixed s, as a sequence in t. By Doob's martingale inequality (see the Theorem in p. 137 of Williams (1991)),

$$\mathbb{P}\left(\sup_{s \leq t < \infty} \| W_{s,t} \| \geq \epsilon \right) \leq \sum_{i=s}^{\infty} \mathbb{E}\left[\| c_i \left(D_i(\theta_i, \lambda_i) - d_i \right) \|^2 \right] / \epsilon^2 \leq \sum_{i=s}^{\infty} c_i^2 \sigma^2 / \epsilon^2 \to 0 \quad (3.39)$$

as $s \to \infty$. So we conclude that

$$\lim_{m \to \infty} \sup_{t_m \leq t < \infty} \| W_{t_m,t} \| I_{\{t_m < \infty\}} = \lim_{m \to \infty} \sup_{t_m \leq t < \infty} \| \sum_{i=t_m}^{t} c_i \left(D_i(\theta_i, \lambda_i) - d_i \right) \| I_{\{t_m < \infty\}} \quad (3.40)$$

a.s. because otherwise there is a non-null set $\mathcal{A}$ on which $t_m < \infty$ infinitely often and

$$\sup_{t_m \leq t < \infty} \| W_{t_m,t} \| I_{\{t_m < \infty\}} \geq \epsilon$$

for infinitely many m, so $\mathbb{P}\left(\sup_{s \leq t < \infty} \| W_{s,t} \| \geq \epsilon \right) \geq \mathbb{P}(\mathcal{A}) > 0$ for infinitely many s, which contradicts (3.39).

Now we prove (3.38). Let C_3 be a constant such that $\| d_n \| \leq C_3$ for $(\theta_n, \lambda_n) \in \mathcal{B}$. When $t_m < \infty$,

$$\begin{aligned}
\sum_{n=t_m}^{t_m^+ - 1} c_n &\geq \| \sum_{n=t_m}^{t_m^+ - 1} c_n d_n \| / C_3 \\
&= \| \sum_{n=t_m}^{t_m^+ - 1} c_n \left(D_n(\theta_n, \lambda_n) + d_n - D_n(\theta_n, \lambda_n) \right) \| / C_3 \\
&= \| \sum_{n=t_m}^{t_m^+ - 1} c_n D_n - W_{t_m, t_m^+ - 1} \| / C_3 \\
&= \| \theta_{t_m^+} - \theta_{t_m} - W_{t_m, t_m^+ - 1} \| / C_3 \\
&\geq \| \theta_{t_m^+} - \theta_{t_m} \| / C_3 - \| W_{t_m, t_m^+ - 1} \| / C_3.
\end{aligned}$$

Since $\|W_{t_m,t_m^+-1}\|I_{\{t_m<\infty\}}/C_3 \to 0$ a.s. as $m \to \infty$ (by (3.40)) and $\|\theta_{t_m^+} - \theta_{t_m}\| \geq \epsilon$, for m sufficiently large, we have

$$\sum_{n=t_m}^{t_m^+-1} c_n \quad \geq \quad \epsilon/(2C_3)$$

and hence

$$\liminf_{m\to\infty} I_{\{t_m<\infty\}} \sum_{n=t_m}^{t_m^+-1} c_n \quad \geq \quad (\epsilon/(2C_3)) \liminf_{m\to\infty} I_{\{t_m<\infty\}},$$

proving (3.38).

Chapter 4

Convex Regression

4.1. Overview

In this chapter, we aim to study the problem of estimating a multivariate regression function under a certain shape restriction such as convexity. This problem is usually referred to as convex regression in the literature. This chapter is concerned with providing a numerically efficient way of computing the best fit of a convex function and proving the consistency of the proposed estimator. We are interested in estimating the unknown function $f_* : [0,1]^d \to \mathbb{R}$ from the observed data $(X_1, Y_1), \ldots, (X_n, Y_n)$, where

$$Y_i = f_*(X_i) + \varepsilon_i$$

for $i \geq 1$, the X_is are continuous $[0,1]^d$–valued independent and identically distributed (iid) random vectors, and the ε_is are iid random variables with zero median and $\mathbb{E}(|\varepsilon_1|) < \infty$.

When f_* is known to be convex, a natural way to estimate f_* is to minimize the sum of squares

$$\psi_n(g) \triangleq \frac{1}{n}\sum_{i=1}^{n}(Y_i - g(X_i))^2$$

or the sum of absolute deviations

$$\varphi_n(g) \triangleq \frac{1}{n}\sum_{i=1}^{n}|Y_i - g(X_i)|$$

over the set of convex functions

$$\mathcal{C} = \left\{g : [0,1]^d \to \mathbb{R} \text{ such that } g \text{ is convex}\right\}.$$

When ψ_n is used as a goodness-of-fit criterion, the fitted function is referred to as "least squared errors" (LSE) estimator (Hildreth, 1954). The LSE-based convex regression in one dimension setting ($d = 1$) is well studied both theoretically and computationally. However, when it comes to multiple dimensions setting, less literatures are available. One important issue is that the LSE estimator suffers from computational inefficiency. Minimization of ψ_n over $\mathcal{C}$ can be formulated as a QP with $(d+1)n$ decision variables and n^2 constraints (Kuosmanen, 2008). The computational burden of solving this QP becomes heavy especially when dn exceeds a few hundred (Lim, 2010). Thus, there is a growing need of fitting a convex function to large-scale data.

To overcome the computational inefficiency of the convex regression estimator, we propose to use φ_n instead of ψ_n as a goodness-of-fit criterion. Using φ_n may be beneficial from a computational point of view because minimization of φ_n over $\mathcal{C}$ can be formulated as an LP rather than a QP. Another advantage of using φ_n is that the least absolute deviations estimators can provide more robust results because they are not sensitive to outliers in the dataset (Bassett and Koenker, 1978; Wagner, 1959).

In this chapter, we use φ_n instead of ψ_n as a goodness-of-fit criterion and investigate the least absolute deviations (LAD) estimator $\hat{g}_n$, which is the minimizer of φ_n over $\mathcal{C}$. We observe that $\hat{g}_n$ can be computed by solving an LP and the LP has a dual problem that can be solved more efficiently. We further discover that the dual problem has a block-angular form in its constraints, and hence, allows application

of decomposition techniques such as Dantzig-Wolfe decomposition. Dantzig-Wolfe decomposition then enables one to compute $\hat{g}_n$ for large-scale data. Our numerical reaults in Section 4.2 show that $\hat{g}_n$ can be computed for a dataset that contains more than $10,000$ datapoints when $d = 1$ while the least squares estimator can only be computed for a dataset containing a few hundred data points. In most of our numerical examples, $\hat{g}_n$ was computed much faster than the least squares estimators.

We also establish the consistency of $\hat{g}_n$ and the derivative $\hat{g}_n$ (when it exists) by proving that $\hat{g}_n$ and the derivative of $\hat{g}_n$ converge to the true values a.s. as n increases to infinity and that this convergence is uniform over any compact subset of $(0,1)^d$.

This chapter is organized as follows. In Section 4.2, we introduce some definitions. Section 4.3 introduces the mathematical framework for our analysis, and precisely states the main theorems (Theorems 3 and 4) of this paper. In Section 4.3, we provide a numerically efficient LP formulation for computing $\hat{g}_n$ while Section 4.5 discusses the numerical behavior of the least absolute deviations estimator compared to that of the least squares estimator. Proofs of the main results are provided in Section 4.6.

4.2. Definitions

For $x \in \mathbb{R}^d$, we write its kth component as x^k, so $x = (x^1, \ldots, x^d)$. We view $x \in \mathbb{R}^d$ as a column vector. We let $\|x\|_\infty = \max(|x^i| : 1 \leq i \leq d)$ and $\|x\| = \left((x^1)^2 + \cdots + (x^d)^2\right)^{1/2}$. For $y \in \mathbb{R}$, we write $y^+ = \max(0, y)$.

For a function $g : [0,1]^d \to \mathbb{R}$, g is differentiable at $x \in (0,1)^d$ if and only if there exists a vector $v \in \mathbb{R}^d$ with the property that

$$\lim_{z \to x} (g(z) - g(x) - v^T(z - x))/\|z - x\| = 0.$$

Such a v, if it exists, is called the gradient of g at x and is denoted by $\nabla g(x)$.

For any convex function $g : [0,1]^d \to \mathbb{R}$, a vector $\xi \in \mathbb{R}^d$ is said to be a subgradient of g at $x \in (0,1)^d$ if $g(y) \geq g(x) + \xi^T(y-x)$ for all $y \in (0,1)^d$. The set of all subgradients of g at x is called the subdifferential of g at x and is denote by $\partial g(x)$. The subdifferential $\partial g(x)$ of a convex function $g : [0,1]^d \to \mathbb{R}$ is non-empty for any $x \in (0,1)^d$; see pp. 215–217 of Rockafellar (1970).

Let $(a_n : n \geq 1)$ and $(b_n : n \geq 1)$ be sequences of real numbers. We say $a_n = O(b_n)$ if there exist positive constants c and n_0 such that $|a_n| \leq c|b_n|$ for all $n \geq n_0$.

4.3. The Main Results

We assume that we observe n pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$, in which

$$Y_i = f_*(X_i) + \varepsilon_i$$

for $i \geq 1$, the X_is are continuous $[0,1]^d$–valued iid random vectors, and the ε_is are iid random variables with zero median and $\mathbb{E}(|\varepsilon_1|) < \infty$.

When f_* is known to be convex, a natural way of estimating it from data is to minimize the sum of absolute deviations

$$\varphi_n(g) \;=\; \frac{1}{n}\sum_{i=1}^{n} |Y_i - g(X_i)|$$

over the set of convex functions $\mathcal{C} = \big\{g : [0,1]^d \to \mathbb{R} \text{ such that } g \text{ is convex}\big\}$. Since there are infinitely many convex functions, this minimization may appear to be computationally intractable. However, the following proposition reveals that this minimization can be formulated as an LP with $(d+3)n$ decision variables and $n^2 + 3n$ constraints.

Proposition 4. *Consider the minimization problem in the decision variables* (g_1, ξ_1),

$\ldots, (g_n, \xi_n)$

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n} |Y_i - g_i| \\ s/t \quad & g_j \geq g_i + \xi_i^T (X_j - X_i), \quad 1 \leq i, j \leq n, \end{aligned} \tag{4.1}$$

where $g_i \in \mathbb{R}$ *and* $\xi_i \in \mathbb{R}^d$ *for* $1 \leq i \leq n$. *Then, the problem (4.1) has a minimizer* $(\hat{g}_1, \hat{\xi}_1), \ldots, (\hat{g}_n, \hat{\xi}_n)$ *and* $\hat{g}_n : [0,1]^d \to \mathbb{R}$, *defined by*

$$\hat{g}_n(x) \;=\; \max_{1 \leq i \leq n} (\hat{g}_i + \hat{\xi}_i^T (x - X_i)) \tag{4.2}$$

for $x \in [0,1]^d$, *minimizes* φ_n *over* $\mathcal{C}$.

Furthermore, the problem (4.1) has a minimizer $(\hat{g}_1, \hat{\xi}_1), \ldots, (\hat{g}_n, \hat{\xi}_n)$ *if and only if* $(\hat{g}_1, (Y_1 - \hat{g}_1)^+, (-Y_1 + \hat{g}_1)^+, \hat{\xi}_1), \ldots, (\hat{g}_n, (Y_n - \hat{g}_n)^+, (-Y_n + \hat{g}_n)^+, \hat{\xi}_n)$ *is a solution to the following LP in the decision variables* $(g_1, p_1, m_1, \xi_1), \ldots, (g_n, p_n, m_n, \xi_n)$:

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n} (p_i + m_i) \\ s/t \quad & g_j \geq g_i + \xi_i^T (X_j - X_i), && 1 \leq i, j \leq n \\ & Y_i - g_i = p_i - m_i, && 1 \leq i \leq n \\ & p_i, m_i \geq 0, && 1 \leq i \leq n, \end{aligned}$$

where $g_i \in \mathbb{R}, p_i \in \mathbb{R}, m_i \in \mathbb{R},$ *and* $\xi_i \in \mathbb{R}^d$ *for* $1 \leq i \leq n$.

Proof. Let $\mathcal{G}_n = \{(g_1, \ldots, g_n) \in \mathbb{R}^n$ such that there exists a convex function $g : [0,1]^d \to \mathbb{R}$ satisfying $g(X_i) = g_i$ for $1 \leq i \leq n\}$. Then, $\mathcal{G}_n$ is nonempty ($(0, \ldots, 0) \in \mathcal{G}_n$), closed and convex by Lemma 2.3 of Seijo and Sen (2011). Note that φ_n is continuous and coercive (i.e., $|\varphi_n(g_1, \ldots, g_n)| \to \infty$ as $\|(g_1, \ldots, g_n)\| \to \infty$). Thus, φ_n has a minimizer $(\hat{g}_1, \ldots, \hat{g}_n)$ over $\mathcal{G}_n$; see Proposition 7.3.1 and Theorem 7.3.7 in pp. 216 and 217 of Kurdila and Zabarankin (2005). Since $(\hat{g}_1, \ldots, \hat{g}_n) \in \mathcal{G}_n$, there exist vectors $\hat{\xi}_1, \ldots, \hat{\xi}_n$ in $\mathbb{R}^d$ satisfying $\hat{g}_j \geq \hat{g}_i + \hat{\xi}_i^T (X_j - X_i)$ for $1 \leq i, j \leq n$, and hence,

$(\hat{g}_1, \hat{\xi}_1), \ldots, (\hat{g}_n, \hat{\xi}_n)$ is a feasible solution of (4.1). Furthermore, $(\hat{g}_1, \hat{\xi}_1), \ldots, (\hat{g}_n, \hat{\xi}_n)$ becomes a minimizer of (4.1) by p. 337 of Boyd and Vandenberghe (2004). The rest of the proposition follows trivially. □

While Proposition 4 asserts that $(\hat{g}_1, \ldots, \hat{g}_n)$ exists, it should be noted that $(\hat{g}_1, \ldots, \hat{g}_n)$ may not be unique. A simple example that illustrates the non-uniqueness of $(\hat{g}_1, \ldots, \hat{g}_n)$ is the following: When $d = 1, n = 4, (X_1, Y_1) = (0.2, 0), (X_2, Y_2) = (0.4, 1), (X_3, Y_3) = (0.6, 1)$, and $(X_4, Y_4) = (0.8, 0)$, any point from the set

$$\Big\{(\hat{g}_1, \hat{\xi}_1) = (a, b), (\hat{g}_2, \hat{\xi}_2) = (a, 0), (\hat{g}_3, \hat{\xi}_3) = (a, 0), (\hat{g}_4, \hat{\xi}_4) = (a, c) : \\ a \in [0, 1], b \in (-\infty, 0], c \in [0, \infty)\Big\}$$

is a minimizer of (4.1). So, $(\hat{g}_1, \hat{g}_2, \hat{g}_3, \hat{g}_4)$ is not unique.

Throughout this paper, we will work with the set of minimizers of φ_n over $\mathcal{C}$:

$$\mathcal{S}_n = \{g_n \in \mathcal{C} : \varphi_n(g_n) \leq \varphi_n(g) \text{ for all } g \in \mathcal{C}\}$$

for $n \geq 1$. By Proposition 4, $\mathcal{S}_n$ is nonempty for all $n \geq 1$ a.s. and Proposition 4 suggests a way of computing an element $\hat{g}_n$ in $\mathcal{S}_n$ by using (4.1) and (4.2). The convex function $\hat{g}_n$ is our estimator for $f_*(\cdot)$. In order to analyze this estimator, we shall impose some probabilistic assumptions on the (X_i, Y_i)'s. In particular, we require that:

A1. $X_1, X_2, \ldots$ is a sequence of iid $[0, 1]^d$–valued random vectors having a common continuous positive density $\kappa : [0, 1]^d \to \mathbb{R}$.

A2. For $i \geq 1$, $Y_i = f_*(X_i) + \varepsilon_i$, where the ε_is satisfy

$$\mathbb{P}\left(\varepsilon_i \in dy_i, 1 \leq i \leq n | X_1, X_2, \ldots\right) = \prod_{i=1}^{n} F(dy_i | X_i)$$

for some family $(F(\cdot|x) : x \in [0,1]^d)$ of cumulative distribution functions.

A3. $\mathbb{E}\left(|f_*(X_1)| + |\varepsilon_1| + \|X_1\|\right) < \infty$, thereby implying that

$$\mathbb{E}\left(|\varepsilon_1|\,|X_1\right) = \int_{\mathbb{R}} |y| F(dy|X_1) < \infty \quad a.s.$$

A4. For each $x \in [0,1]^d$, we have $F(0|x) = 1/2$.

A5. f_* is bounded; i.e., there exists a positive constant M such that $|f_*(x)| \leq M$ for all $x \in [0,1]^d$.

We are now ready to state our main results.

Theorem 3. *Assume A1–A5 and that $f_* \in \mathcal{C}$. Then for each $0 < c < 1/2$,*

$$\sup_{x \in [c,1-c]^d, \hat{g}_n \in \mathcal{S}_n} |\hat{g}_n(x) - f_*(x)| \to 0 \quad a.s.$$

as $n \to \infty$.

Theorem 4. *Assume A1–A5 and that $f_* \in \mathcal{C}$. If f_* is differentiable at $z \in (0,1)^d$, then*

$$\sup_{\xi \in \partial \hat{g}_n(z), \hat{g}_n \in \mathcal{S}_n} \|\xi - \nabla f_*(z)\| \to 0$$

as $n \to \infty$ a.s.

Furthermore, if f_* is differentiable on $[c, 1-c]^d$ for any $0 < c \leq 1/2$,

$$\sup_{x \in [c,1-c]^d, \xi \in \partial \hat{g}_n(x), \hat{g}_n \in \mathcal{S}_n} \|\xi - \nabla f_*(x)\| \to 0$$

as $n \to \infty$ a.s.

Theorems 3 and 4 justify our choice of the least absolute deviations estimator $\hat{g}_n$ as an estimator of f_*. The next section is concerned with providing an efficient way of computing $\hat{g}_n$.

4.4. A More Efficient LP Formulation for Computing the Proposed Estimator

In this section, we present an efficient LP formulation for computing $\hat{g}_n$. By Proposition 4, $\hat{g}_n$ can be computed by solving the following LP in the decision variables $(g_1, p_1, m_1, \xi_1), \ldots, (g_n, p_n, m_n, \xi_n)$

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n}(p_i + m_i) \\ \text{s/t} \quad & g_j \geq g_i + \xi_i^T(X_j - X_i), \quad 1 \leq i, j \leq n \\ & Y_i - g_i = p_i - m_i, \quad 1 \leq i \leq n \\ & p_i, m_i \geq 0, \quad 1 \leq i \leq n. \end{aligned} \tag{4.3}$$

We notice that the dual problem of (4.3) is the following LP with the decision variables $(s_{ij} : 1 \leq i, j \leq n)$ and $(t_i : 1 \leq i \leq n)$

$$\begin{array}{lccccccccl} \max & Y^T t & & & & & & & & \\ \text{s/t} & A_1 s_1 + & A_2 s_2 + & \cdots + & A_n s_n + & I_n t & = & 0_n & & \\ & b_1^T s_1 & & & & & = & 0 & & \\ & & b_2^T s_2 & & & & = & 0 & & \\ & & & \ddots & & & & \vdots & & \\ & & & & b_n^T s_n & & = & 0 & & \\ & & & & & I_n t & \leq & 1_n & & \\ & & & & & I_n t & \geq & -1_n & & \\ & & & & & s_i & \geq & 0_n, & 1 \leq i \leq n, & \end{array} \tag{4.4}$$

where $Y = (Y_1, \ldots, Y_n)^T$, $t = (t_1, \ldots, t_n)^T \in \mathbb{R}^n$, $s_i = (s_{i1}, \ldots, s_{in})^T \in \mathbb{R}^n$ for $1 \leq i \leq$

n, $A_i = (a_{jk} : 1 \leq j, k \leq n)$ with

$$a_{jk} = \begin{cases} 1, & j = k, j \neq i \\ -1, & j = i, k \neq i \\ 0, & \text{otherwise,} \end{cases}$$

$b_i = (X_i - X_1, \ldots, X_i - X_n)^T$ for $1 \leq i \leq n$, I_n is an n by n identify matrix, 1_n is an n by 1 vector of all ones, and 0_n is an n by 1 vector of all zeros. The s_{ij}s and t_is are dual variables corresponding to the first and second sets of constraints of (4.3), respectively.

The dual problem (4.4) has two sets of decision variables $(s_{ij} : 1 \leq i, j \leq n)$ and $(t_i : 1 \leq i \leq n)$. The two sets of variables are related only through the first constraint in (4.4). Thus, (4.4) has a block structure in its constraints and hence allows application of decomposition techniques such as Dantzig-Wolfe decomposition.

4.4.1 Dantzig-Wolfe Decomposition

The Dantzig-Wolfe decomposition improves the computational efficiency of an LP significantly especially when the LP problem has a nice block-angular structure. Instead of solving the original problem with complicating constraints, two types of problems are solved iteratively, a so-called master problem and a so-called subproblem without complicating constraints. Since the master problems and the subproblems have much less decision variables and constraints than the original LP problem, the memory issue can be solved by repeating to solve a series of small LP problems. In such a way, we can efficiently handle a large-scale LP problem without running out the memory.

4.4.2 Dantzig-Wolfe Decomposition Algorithm for Convex Regression

To apply the Dantzig-Wolfe decomposition, we consider the dual problem (4.4). For $i = 1, \ldots, n$, define $S_i = \{s_i \geq 0 | b_i^T s_i = 0\}$ and $T = \{t | I_n t \leq 1_n \text{ and } I_n t \geq -1_n\}$, then (4.4) becomes

$$\begin{aligned} \max \quad & Y^T t \\ \text{s/t} \quad & \sum_{i=1}^{n} A_i s_i + I_n t = 0, \\ & s_i \in S_i, \quad 1 \leq i \leq n \\ & t \in T. \end{aligned} \tag{4.5}$$

For $i = 1, \ldots, n$, let s_i^k, $k \in K_{s_i}$, be all the extreme points of set S_i. Let $d_{s_i}^l$, $l \in L_{s_i}$, denote all the extreme rays of set S_i. Also let t^h, $h \in H_t$, be all the extreme points of set T. Since T is a bounded set, it does not have any extreme rays. According to the resolution theorem (Theorem 4.15 of Bertsimas and Tsitsiklis (1997)), any solution s_i of S_i can be represented as

$$s_i = \sum_{k \in K_{s_i}} \lambda_{s_i}^k s_i^k + \sum_{l \in L_{s_i}} \mu_{s_i}^l d_{s_i}^l,$$

where $\lambda_{s_i}^k$ and $\mu_{s_i}^l$ are nonnegative and satisfy the convexity constraint

$$\sum_{k \in K_{s_i}} \lambda_{s_i}^k = 1, \quad 1 \leq i \leq n.$$

Follow the same principle, any solution t of T can be written as

$$t = \sum_{h \in H_t} \lambda_t^h t^h,$$

where λ_t^h is nonnegative and satisfy

$$\sum_{h \in H_t} \lambda_t^h = 1.$$

Then (4.5) can be reformulated as

$$\begin{aligned}
\max \quad & \sum_{h \in H_t} \lambda_t^h Y^T t^h && (4.6)\\
\text{s/t} \quad & \sum_{i=1}^{n} \sum_{k \in K_{s_i}} \lambda_{s_i}^k A_i s_i^k + \sum_{i=1}^{n} \sum_{l \in L_{s_i}} \mu_{s_i}^l A_i d_{s_i}^l + \sum_{h \in H_t} \lambda_t^h I_n t^h = 0, && (4.7)\\
& \sum_{k \in K_{s_i}} \lambda_{s_i}^k = 1, \quad 1 \le i \le n && (4.8)\\
& \sum_{h \in H_t} \lambda_t^h = 1, && (4.9)\\
& \lambda_{s_i}^k, \lambda_t^h, \mu_{s_i}^l \ge 0, \quad \forall h, i, k, l.
\end{aligned}$$

This so called master problem is equivalent to the original problem (4.4) and is a standard linear programming problem with decision variables $\lambda_{s_i}^k$, λ_t^h and $\mu_{s_i}^l$. This equivalent formulation only has $2n+1$ equality constraints, where the original one has $(d+3)n$ constraints. But the number of decision variables in this formulation is typically very large. Since the optimal sets K_{s_i}, L_{s_i} and H_t are very large and unknown for us, we can use a delayed column generation scheme to generate them iteration by iteration in order to solve (4.6) more efficient. In this case, the master problem is called a restricted master problem since only a subset of columns associated with each decision variables are included. At each iteration, new columns are generated by solving a series of pricing problems. To decide if a column has potential to improve the current master problem, we evaluate its reduced as we usually do in the revised simplex method. If the reduced cost is positive for our maximization problem, then this new column should be selected to enter into the restricted master problem.

Evaluating the reduced cost explicitly for every $\lambda_{s_i}^k$, λ_t^h and $\mu_{s_i}^l$ is computationally expensive since there are numerous such variables. Instead, we can solve a series of LP problems to achieve this goal. More specifically, we consider the following LP problems:

$$\begin{aligned} \max \quad & (-\pi^T A_i) s_i \\ \text{s/t} \quad & s_i \in S_i, \end{aligned}$$

for $i = 1, \ldots, n$, and

$$\begin{aligned} \max \quad & (Y^T - \pi^T I_n) t \\ \text{s/t} \quad & t \in T, \end{aligned}$$

where π is the optimal dual solution associated with constraint (4.7) of the current master problem. Also let σ_{s_i} and σ_t be the optimal dual solutions associated with constraint (4.8) and (4.9), respectively.

The above LP problems are also called subproblems and can be easily solved by the revised simplex method. When solving such subproblems, there are three possible results.

1. If the subproblem is bounded and the objective function value $(-\pi^T A_i) s_i^k > \sigma_{s_i}$ for some k or $(Y^T - \pi^T I_n) t^h > \sigma_t$ for some h, then the reduced cost of $\lambda_{s_i}^k$ or λ_t^h is positive. We add a new column

$$\left[\begin{array}{c} 0 \\ \hline A_i s_i^k \\ e_i \\ 0 \end{array}\right] \text{ or } \left[\begin{array}{c} Y^T t^h \\ \hline I_n t^h \\ 0 \\ 1 \end{array}\right],$$

   where e_i is an $n \times 1$ unit vector whose ith entry is 1.

2. If the subproblem is unbounded, i.e., the extreme ray $d_{s_i}^l$ returned by the simplex method satisfies $-\pi^T A_i d_{s_i}^l > 0$ for some l, then the reduced cost of $\mu_{s_i}^l$ is positive. So a new column

$$\begin{bmatrix} 0 \\ \hline A_i d_{s_i}^l \\ 0 \\ 0 \end{bmatrix}$$

associated with $\mu_{s_i}^l$ can be entered into the master problem.

3. If for all subproblems, $(-\pi^T A_i) s_i^k \leq \sigma_{s_i}$ and $(Y^T - \pi^T I_n) t^h \leq \sigma_t$, then the reduced cost for each decision variable in restricted master problem is nonpositive. Thus, the overall optimal solution of the original problem has been obtained.

Based on the Dantzig-Wolfe decomposition principle introduced above, the decomposition algorithm on solving our convex regression problem can be summarized as follows.

Algorithm 3: Dantzig-Wolfe Decomposition Algorithm for Convex Regression

---

Step 0. Initialize: Start from the solution $s_i^0 = \{0, \ldots, 0\}$ $(1 \leq i \leq n)$ and $t^0 = \{0, \ldots, 0\}$. Set iteration counter $v = 0$. Let $K_{s_i} = \{s_i^0\}$, $H_t = \{t^0\}$ and $L_{s_i} = \emptyset$.

Step 1. Master Problem Step: Solve the master problem (4.6) to obtain the optimal dual solution π^v, $\sigma_{s_i}^v$ and σ_t^v.

Step 2. Subproblem Step: Solve the subproblem

$$\begin{array}{ll} \max & (-(\pi^v)^T A_i)s_i \\ \text{s/t} & s_i \in S_i, \end{array} \tag{4.10}$$

for $i = 1, \ldots, n$, and

$$\begin{array}{ll} \max & (Y^T - (\pi^v)^T I_n)t \\ \text{s/t} & t \in T. \end{array} \tag{4.11}$$

If the subproblem is bounded and the optimal solution s_i^* or t^* satisfies $(-(\pi^v)^T A_i)s_i^* > \sigma_{s_i}^v$ or $(Y^T - (\pi^v)^T I_n)t^* > \sigma_t^v$, then add a new column

$$\begin{bmatrix} 0 \\ \hline A_i s_i^* \\ e_i \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} Y^T t^* \\ \hline I_n t^* \\ 0 \\ 1 \end{bmatrix},$$

to the current master problem. Let $K_{s_i} = K_{s_i} \cup \{s_i^*\}$ or $H_t = H_t \cup \{t^*\}$.

If the subproblem is unbounded, i.e., the extreme ray $d_{s_i}^*$ satisfies $-(\pi^v)^T A_i d_{s_i}^* > 0$, then add a new column

$$\begin{bmatrix} 0 \\ \hline A_i d_{s_i}^* \\ 0 \\ 0 \end{bmatrix}$$

to the current master problem. Let $L_{s_i} = L_{s_i} \cup \{d_{s_i}^*\}$.

Step 3. Optimality Check: If for all subproblems, $(-(\pi^v)^T A_i)s_i^* \leq \sigma_{s_i}^v$ and $(Y^T - (\pi^v)^T I_n)t^* \leq \sigma_t^h$, then the solution of the current master problem is the

overall optimal solution for the original problem; otherwise, let $v = v + 1$, go to Step 1.

Step 4. Obtain the Estimator: The least absolute deviations estimator $\hat{g}_n = \pi^v$.

The Dantzig-Wolfe decomposition algorithm has been proved to terminate after a finite number of iterations (Dantzig and Wolfe, 1961).

The reason that formulation (4.4) can be solved more efficiently than formulation (4.3) is two-fold. When solving an LP problem using the simplex method, it is more efficient to solve the dual problem than the primal problem if the primal problem has much more constraints than the decision variables; see p. 147 of Bradley et al. (1977) and p. 234 of Grover (2004) for details. It is the case with the primal problem (4.3) and the dual problem (4.4) because (4.3) has $O(n)$ decision variables and $O(n^2)$ constraints while (4.4) has $O(n^2)$ decision variables and $O(n)$ constraints. Second, one can apply Dantzig-Wolfe decomposition to solve (4.4) and thus can compute $\hat{g}_n$ for larger datasets than (4.3) can handle. Note that in Step 2 of this procedure, the subproblem for s_i, $1 \leq i \leq n$, can be easily solved by the revised simplex method. And the memory requirement for this type of subproblem is $O(d^2)$, which is the size of the revised simplex tableau. The subproblem for t has a straightforward optimal solution since t_j, $1 \leq j \leq n$, achieves the optimality at either 1 or -1, depending on the corresponding coefficient in the objective function. So the memory requirement for this type of subproblem is a constant. In addition, the master problem has $2n + 1$ equality constraints, which means the revised simplex algorithm will require $O((2n + 1)^2)$ memory space. At each iteration, only one simplex tableau is maintained for the master problem and one for the subproblem. Hence, the decomposition algorithm requires much less physical memory to compute the estimator and might solve convex regression problem with very large n.

4.4.3 Computing Strategies

The Dantzig-Wolfe decomposition algorithms is known to converge very slow at the later iterations of the process. Here we give several possible ways to accelerate the algorithm on computing our convex estimator $\hat{g}_n$.

4.4.3.1 Strategies to enter new columns

In order to improve the computational efficiency of the Dantzig-Wolfe decomposition algorithm, several strategies to enter new columns to the mater problem have been proposed. One commonly used strategy is choosing the column with the most positive reduced cost at each iteration. Although this strategy works for small-size problems, it doesn't perform well when n is large because after solving $n+1$ subproblems with a lot of computation time only one candidate is entered into the master problem. Another strategy is entering as many of the columns provided by the subproblems as possible into the master problem; that is, for each subproblem, if the reduced cost is positive then add the corresponding column to the master problem. This strategy might make the algorithm converge more quickly since no information provided by the subproblems is wasted. But the drawback is that the size of the master problem will increase much faster and the simplex method might spend more time to solve the master problem at each iteration. Our suggested strategy is forming several groups for the first n subproblems and entering the column with the most negative positive cost in each group. In this way, we can control the size of the master problem without losing too much information provided by the subproblems.

4.4.3.2 Reduce the number of convexity constraints

In the master problem (4.6), there are $n+1$ convexity constraints((4.8) and (4.9)). However, we can reduce the number of convexity constraints to 1 by viewing the $n+1$ blocks in (4.6) as a single block. Then at each iteration of the decomposition

algorithm, we only need to solve one subproblem

$$\begin{aligned} \max \quad & (-(\pi^v)^T A_0)s_0 + \ldots + (-(\pi^v)^T A_n)s_n + (Y^T - (\pi^v)^T I_n)t \\ \text{s/t} \quad & s_i \in S_i, \quad 1 \leq i \leq n \\ & t \in T. \end{aligned} \tag{4.12}$$

According to Ho and Loute (1981), every solution of the subproblem with positive reduced cost has the potential to improve the master problem. Hence, we can generate multiple columns from subproblem (4.12) as long as the columns generated have positive reduced costs. We noticed that subproblem (4.12) is a separable maximization problem since decision variables s_i and t have separable coefficients and constraints. So we can generate a solution of (4.12) by solving a subproblem (4.10) or (4.11) in Step 2 of our decomposition algorithm. Suppose that s_i^* is an optimal solution of (4.10) and satisfies $(-(\pi^v)^T A_i)s_i^* > \sigma_{s_i}^v$, then the solution $(0, \ldots, (s_i^*)^T, \ldots, 0)^T$ is a solution of (4.12) with positive reduced cost. Then we can generate a column based on this solution s_i^* and enter it into the master problem. The same principle applies to t and d_{s_i}. By using this alternative formulation, we are able to reduce the total number of constraints of the master problem to $n + 1$ and solve the master problem much more efficient when n is very large.

4.4.3.3 Stabilization techniques for Dantzig-Wolfe decomposition

When applying the Dantzig-Wolfe decomposition method to (4.4), we observe that although the algorithm moves to a near optimal solution very fast, it makes little process per iteration towards the optimum. This poor convergence is known as tailing-off effect of column generation (Lï¿½bbecke and Desrosiers, 2005). The reason of this phenomenon is that the dual solution does not converge smoothly, but oscillates around the optimum. One simple treatment for this issue is imposing lower and upper bounds for the dual variables so that the new dual solution is forced to lie

in the neighborhood of the optimal dual solution of the current restricted master problem. This method is usually called Boxstep and was introduced by Marsten et al. (1975). In order to accelerate the Dantzig-Wolfe decomposition method to solve (4.4), we implement a variant of the Boxstep which uses the linear programming frame work and combines perturbations and penalties to stabilize the column generation (du Merle et al., 1999). To constrain the dual variables in the dual space, we consider an augmented master problem

$$
\begin{aligned}
\max \quad & \sum_{h\in H_t} \lambda_t^h Y^T t^h + \delta_+^T y_+ - \delta_-^T y_- \\
\text{s/t} \quad & \sum_{i=1}^{n} \sum_{k\in K_{s_i}} \lambda_{s_i}^k A_i s_i^k + \sum_{i=1}^{n} \sum_{l\in L_{s_i}} \mu_{s_i}^l A_i d_{s_i}^l + \sum_{h\in H_t} \lambda_t^h I_n t^h + I_n y_+ - I_n y_- = 0, \\
& \sum_{k\in K_{s_i}} \lambda_{s_i}^k = 1, \quad 1 \le i \le n \\
& \sum_{h\in H_t} \lambda_t^h = 1, \\
& y_+ \le \epsilon_+ \\
& y_- \le \epsilon_- \\
& \lambda_{s_i}^k, \lambda_t^h, \mu_{s_i}^l \ge 0, \quad \forall h, i, k, l \\
& y_+, y_- \ge 0,
\end{aligned}
$$

where y_+ and y_- are vectors with upper bounds ϵ_+ and ϵ_-, respectively. As a result, the dual variables π are constrained by $\delta_- - z_- \le \pi \le \delta_+ + z_+$ in the corresponding dual problem. In the above box constraint, z_+ and z_- decide the amounts to penalize if dual variables π locate outside of the box $[\delta_-, \delta_+]$. However, to make sure the augmented master problem has the same solution with the original one, we have to let $y_+ = y_- = 0$ in the final iteration. This goal can be achieved by letting 1)$\epsilon_+ = \epsilon_- = 0$ or 2) $\delta_- \le \pi \le \delta_+$. To control the dual variables' variation, two possible strategies are used to update parameters ϵ_+, ϵ_-, δ_+ and δ_- dynamically. If

the new dual solution π^v obtained at iteration v is outside of the box $[\delta_-, \delta_+]$, we recenter the box at π^v and increase the width of the box. This can be done by setting δ_+ and δ_- to the new dual solution π^v and decreasing the penalty values ϵ_+ and ϵ_-. On the other hand, if the new dual solution π^v is inside the box defined by $[\delta_-, \delta_+]$, we recenter the box at π^v and decrease the width of the box by letting $\delta_+ = \delta_- = \pi^v$ and increasing the penalty values ϵ_+ and ϵ_-. In addition, to reduce the uncertainty of estimating stabilization center by π^h at each iteration, we can incorporate a positive perturbation parameter ξ and let $\delta_+ + \xi = \delta_- - \xi = \pi^v$.

In the next section, we compare formulations (4.3) and (4.4) through numerical examples.

4.5. Numerical Results

In this section, we investigate how fast $\hat{g}_n$ can be computed by solving (4.4) through common LP solving techniques such as the simplex method and the interior point method. We further illustrate how $\hat{g}_n$ can be computed for large datasets by solving (4.4) with Dantzig-Wolfe decomposition.

We are particularly interested in the relative performance of $\hat{g}_n$ compared to that of the least squares estimator. The least squares estimator is defined as the minimizing values $\tilde{g}_n(X_1), \ldots, \tilde{g}_n(X_n)$ of the following QP in the decision variables $(g_1, \xi_1), \ldots, (g_n, \xi_n)$

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n}(Y_i - g_i)^2 \\ \text{s/t} \quad & g_j \geq g_i + \xi_i^T(X_j - X_i), \quad 1 \leq i, j \leq n; \end{aligned} \tag{4.13}$$

see Lim and Glynn (2012) for details.

In Section 4.5.1, we observe numerically how $\hat{g}_n$ converges to the true value as n

increases to infinity. The performance of $\hat{g}_n$ is compared to that of the least squares estimator $\tilde{g}_n(X_1), \ldots, \tilde{g}_n(X_n)$.

In Section 4.5.2, we compare formulations (4.4) and (4.13) in three numerical examples: 1) a stylized model, 2) an inventory control system, and 3) a tandem queueing network. In each of these examples, we consider four different computational strategies. Estimator 1 is the least squares estimator $\tilde{g}_n(X_1), \ldots, \tilde{g}_n(X_n)$. It is computed by solving (4.13) through the interior point method in CPLEX. Estimator 2 is $\hat{g}_n$ and is computed by solving (4.4) through the simplex method in CPLEX. Estimator 3 is $\hat{g}_n$ and is computed by solving (4.4) through the interior point method in CPLEX. To compute estimator 4, we implement the Dantzig-Wolfe decomposition algorithm we summarized in Section 4.4.1 and include those computing strategies in Section 4.4.3 to improve the performance. To implement the stabilized column generation technique, we set δ_+^v and δ_-^v around π^v at the first l iterations ($v \leq l$) of the algorithm. After l iterations ($v > l$), we update δ_+^v and δ_-^v if π^v is the best dual solution found so far, where the quality of π^v can be estimated through the lower bound of the restricted master problem. In addition, we decrease ϵ_+^v and ϵ_-^v by a factor of 2 if no column can be entered into the master problem.

All the numerical experiments are conducted on a computer with a processor of 2.33 GHz and a RAM of 12 GB.

4.5.1 Consistency

4.5.1.1 One-dimensional case

We consider the case where $f_*(x) = (x - 0.5)^2$ for $x \in [0, 1]$, $X_i = i/n$ for $1 \leq i \leq n$, and ε_i follows $\log N(-2, 2)$ with probability 0.5 and $-\log N(-2, 2)$ with probability 0.5, where $\log N(-2, 2)$ is a lognormal distribution with normal mean -2 and variance 2. Using (X_i, Y_i) for $1 \leq i \leq n$, we compute $\hat{g}_n(X_i)$ and the $\tilde{g}_n(X_i)$s. Table 4.1 reports

the averages (Mean) and the standard deviation (Std) of

$$d_1(\hat{g}_n, f_*) \triangleq \sup_{X_i \in [0.2,0.8]} |\hat{g}_n(X_i) - f_*(X_i)|$$

and the averages (Mean) and the standard deviation (Std) of

$$d_1(\tilde{g}_n, f_*) \triangleq \sup_{X_i \in [0.2,0.8]} |\tilde{g}_n(X_i) - f_*(X_i)|,$$

based on 100 iid replications for each value of n.

Table 4.1: Consistency: One-dimensional case

	$d_1(\hat{g}_n, f_*)$		$d_1(\tilde{g}_n, f_*)$	
n	Mean	Std	Mean	Std
10	0.31	0.38	0.54	0.54
20	0.16	0.09	0.36	0.28
50	0.10	0.05	0.28	0.23
100	0.06	0.03	0.20	0.13
150	0.06	0.02	0.18	0.10
200	0.05	0.02	0.17	0.11
400	0.04	0.01	0.11	0.06

4.5.1.2 Two-dimensional case

We consider the case where $f_*(x) = (x^1 - 0.5)^2 + (x^2 - 1.0)^2$ for $x = (x^1, x^2) \in [0,1]^2$, $X_{ij} = (i/n^{1/2}, j/n^{1/2})$ for $1 \le i, j \le n^{1/2}$, and the noisy measurement Y_{ij} at X_{ij} follows $f_*(X_{ij}) + \log N(-2,2)$ with probability 0.5 and $f_*(X_{ij}) - \log N(-2,2)$ with probability 0.5, where $\log N(-2,2)$ is a lognormal distribution with normal mean -2 and variance 2. Using (X_{ij}, Y_{ij}) for $1 \le i, j \le n^{1/2}$, we compute $\hat{g}_n(X_{ij})$ and the $\tilde{g}_n(X_{ij})$s. Table 4.2 reports the averages (Mean) and the standard deviation (Std) of

$$d_2(\hat{g}_n, f_*) \triangleq \sup_{X_{ij} \in [0.2,0.8]^2} |\hat{g}_n(X_{ij}) - f_*(X_{ij})|$$

and the averages (Mean) and the standard deviation (Std) of

$$d_2(\tilde{g}_n, f_*) \triangleq \sup_{X_{ij}\in[0.2,0.8]^2} |\tilde{g}_n(X_{ij}) - f_*(X_{ij})|,$$

based on 100 iid replications for each value of n.

Table 4.2: Consistency: Two-dimensional case

	$d_2(\hat{g}_n, f_*)$		$d_2(\tilde{g}_n, f_*)$	
n	Mean	Std	Mean	Std
16	0.63	1.01	0.89	1.17
64	0.16	0.07	0.54	0.39
144	0.11	0.04	0.38	0.24
225	0.10	0.03	0.33	0.27
400	0.03	0.01	0.12	0.07

In the above examples, the proposed estimator displays good performance.

4.5.2 Time Required to Compute Estimators 1, 2, 3, and 4

4.5.2.1 One-dimensional case: a stylized model

We consider the case where $f_* : [0,1] \to \mathbb{R}$ is defined by $f_*(x) = (x - 0.5)^2$ for $x \in [0,1], X_i = i/n$ for $1 \le i \le n$, and ε_i is normally distributed with mean zero and variance 0.05^2 for $1 \le i \le n$. Using (X_i, Y_i) for $1 \le i \le n$, we compute estimators 1, 2, 3, and 4. The parameters used to stabilize the Dantzig-Wolfe decomposition method are: $l = 200$, $\epsilon_+^0 = \epsilon_-^0 = Y$, $\xi = 0.002$ for the first 150 iterations and $\xi = 0.01$ for the rest.

Table 4.3 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory could not execute the procedure.

Table 4.3: Performance of estimators 1, 2, 3, and 4 for a quadratic function

	CPU Time (sec)							
	Estimator 1		Estimator 2		Estimator 3		Estimator 4	
n	Mean	Std	Mean	Std	Mean	Std	Mean	Std
5	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.01
50	2.00	0.09	0.05	0.01	0.09	0.03	0.07	0.01
400	53.73	5.47	5.85	0.16	15.16	0.96	2.27	0.24
600	-	-	18.65	0.58	55.24	2.14	5.28	0.47
1000	-	-	41.67	1.29	264.97	11.12	15.32	1.45
1400	-	-	114.79	5.71	681.80	21.48	32.21	3.35
1600	-	-	-	-	-	-	44.76	4.75
2000	-	-	-	-	-	-	71.36	7.20
5000	-	-	-	-	-	-	679.40	74.05
10000	-	-	-	-	-	-	4177.68	532.36

4.5.2.2 Two-dimensional case: (Q, r) inventory system

We consider a single–item continuous–review (Q, r) inventory system, where we place an order with a fixed quantity Q whenever the inventory position (= on hand stock minus backorders plus any outstanding orders) drops below a prespecified quantity r. The replenishment lead time is assumed to be one unit of time. When an order is placed, a fixed setup cost of \$100 is incurred. A holding cost of \$10 or a penalty cost of \$25 per unit time is charged against any inventory or backorder. We further assume that demand follows a Poisson process with a rate of 50 per unit time. Any unfilled demand is backordered. Our goal is to estimate the steady-state mean total costs per unit time $C(Q, r)$, which is proven to be jointly convex in Q and r (p. 89 of (Zheng, 1992)). To compute $C(Q, r)$, we select the values for (Q, r) at $X_{ij} = (35 + 10i/(n^{1/2}), 35 + 10j/(n^{1/2}))$ for $1 \leq i, j \leq n^{1/2}$, simulate the inventory system up to time 100 at each X_{ij}, compute the average Y_{ij} of all the costs up to time 100 at each X_{ij}, and obtain the average of 20 independent copies of Y_{ij}, say $\overline{Y}_{ij}$. Using $(X_{ij}, \overline{Y}_{ij})$ for $1 \leq i, j \leq n^{1/2}$, we compute estimators 1, 2, 3, and 4. The parameters

used to stabilize the Dantzig-Wolfe decomposition method are: $l = 200$, $\epsilon_+^0 = \epsilon_-^0 = Y$, $\xi = 0.005$ for the first 100 iterations and $\xi = 0.01$ for the rest.

Table 4.4 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory could not execute the procedure.

Table 4.4: Performance of estimators 1, 2, 3, and 4 for a (Q, r) inventory system

	CPU Time (sec)							
	Estimator 1		Estimator 2		Estimator 3		Estimator 4	
n	Mean	Std	Mean	Std	Mean	Std	Mean	Std
64	1.98	0.15	0.19	0.02	0.10	0.02	0.48	0.08
100	3.29	0.19	0.32	0.02	0.25	0.03	1.80	0.31
625	-	-	56.61	1.72	62.28	1.38	34.39	4.71
1600	-	-	-	-	-	-	265.30	42.47
2500	-	-	-	-	-	-	733.65	91.31
6400	-	-	-	-	-	-	12244.76	1701.47

4.5.2.3 Three-dimensional case: tandem queue

We consider a queueing system of three single-server stations connected in tandem, where the interarrival times follow a uniform distribution over $[2.43, 3.43]$ and the service times at server i follow a uniform distribution over $[x_i - 0.5, x_i + 0.5]$ for $1 \leq i \leq 3$. The interarrival times and service times are independent of each other and the first in/first out queueing discipline is used at each server. Each server has unlimited buffer space. We wish to compute the expected sojourn time $s_{600}(x_1, x_2, x_3)$ of the 600th customer. Even though there is no explicit formula for s_{600}, the convexity of s_{600} has been proven; see p. 141 of Shanthikumar and Yao (1991) for details. To compute s_{600}, we simulate the tandem queue at $X_{ijk} = (2.85 + 0.06i/(n^{1/3}), 2.85 + 0.06j/(n^{1/3}), 2.85 + 0.06k/(n^{1/3}))$ for $1 \leq i, j, k \leq n^{1/3}$ and compute the sojourn time

Y_{ijk} of the 600th customer. We then obtain the average of 30 independent copies of Y_{ijk}, say $\overline{Y}_{ijk}$. Using $(X_{ijk}, \overline{Y}_{ijk})$ for $1 \leq i, j, k \leq n^{1/3}$, we compute estimators 1, 2, 3, and 4. The parameters used to stabilize the Dantzig-Wolfe decomposition method are: $l = 200$, $\epsilon_+^0 = \epsilon_-^0 = Y$, $\xi = 0.005$ for the first 100 iterations and $\xi = 0.01$ for the rest.

Table 4.5 reports the averages (Mean) and the standard deviation (Std), based on 30 independent copies, of the CPU time required to compute estimators 1, 2, 3, and 4. The symbol - means that the computer ran out of memory could not execute the procedure.

Table 4.5: Performance of estimators 1, 2, 3, and 4 for a tandem queue

	CPU Time (sec)							
	Estimator 1		Estimator 2		Estimator 3		Estimator 4	
n	Mean	Std	Mean	Std	Mean	Std	Mean	Std
64	0.18	0.04	0.06	0.01	0.07	0.01	0.38	0.06
216	50.06	47.92	2.09	0.09	1.09	0.08	7.51	0.87
512	-	-	51.22	2.43	15.81	1.51	27.51	4.50
1000	-	-	340.89	19.86	187.18	16.33	121.81	24.29
1728	-	-	-	-	-	-	862.55	127.83
4096	-	-	-	-	-	-	11410.52	2895.74

In each of the three examples, the proposed estimator is computed faster and for larger datasets than the least squares estimator.

4.6. Proofs of Theorems 3 and 4

This section is devoted to supplying the details of the proofs of Theorems 3 and 4. We first prove Theorem 3 and follow that with a proof of Theorem 4.

4.6.1 Proof of Theorem 3

Our proof of Theorem 3 can be broken down into a number of key steps.

Step 1 Since $\varphi_n(\hat{g}_n) \leq \varphi_n(f_*)$ for any $\hat{g}_n \in \mathcal{S}_n$, we must have

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{g}_n(X_i)| &\leq \frac{1}{n}\sum_{i=1}^{n} |Y_i - f_*(X_i)| \\ &= \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i) + \varepsilon_i - f_*(X_i)| = \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| \,. \end{aligned} \tag{4.14}$$

Step 2 Observe that, for any $\hat{g}_n \in \mathcal{S}_n$, we must have

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i)| &\leq \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| + \frac{1}{n}\sum_{i=1}^{n} |Y_i| \quad \text{by (4.14)} \\ &\leq \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| + \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i) + \varepsilon_i| \\ &\leq \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| + \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i)| + \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| \,. \end{aligned}$$

So,

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i)| \leq \quad 2\mathbb{E}\,|\varepsilon_1| + \mathbb{E}\,|f_*(X_1)| + 1 \triangleq \beta < \infty$$

a.s. for n sufficiently large by A3 and the strong law of large numbers.

Step 3 We show that for any $A \subset [0,1]^d$ with a nonempty interior, there exists $\tilde{\beta}(A)$ such that

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in A} |\hat{g}_n(x) - f_*(x)| \leq \tilde{\beta}(A)$$

a.s. for n sufficiently large.

To fill in the details, we observe that the strong law of large numbers and A3 ensure

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i)| &= \frac{1}{n}\sum_{i=1}^{n} |Y_i - \varepsilon_i| \\ &\leq \frac{1}{n}\sum_{i=1}^{n} |Y_i| + \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i| \leq \mathbb{E}\,|Y_1| + \mathbb{E}\,|\varepsilon_1| + 1 \triangleq \tilde{\beta} \end{aligned}$$

a.s. for n sufficiently large.

The strong law of large numbers also guarantees that

$$\liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A) \geq \mathbb{P}(X_1 \in A)$$

a.s.

Let

$$\begin{aligned} B \;=\; \Bigg\{ & \sup_{\hat{g}_n \in \mathcal{S}_n} \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i)| \leq \beta \text{ for } n \text{ sufficiently large}, \\ & \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i)| \leq \tilde{\beta} \text{ for } n \text{ sufficiently large}, \\ & \text{and } \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A) \geq \mathbb{P}(X_1 \in A) \Bigg\}, \end{aligned}$$

then by Step 2 and the above arguments, we have $\mathbb{P}(B) = 1$.

Set $\tilde{\beta}(A) \triangleq (\beta + \tilde{\beta} + 1)/\mathbb{P}(X_1 \in A)$. We will prove that $\mathbb{P}(C) = 1$, where

$$C = \left\{ \sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in A} |\hat{g}_n(x) - f_*(x)| \leq \tilde{\beta}(A) \text{ for } n \text{ sufficiently large} \right\},$$

by showing that $B \cap C^c = \emptyset$.

Suppose, on the contrary, that $\omega \in B \cap C^c$. Then for such ω, there exists $\hat{g}_n \in \mathcal{S}_n$ such that

$$\inf_{x \in A} |\hat{g}_n(x) - f_*(x)| > \tilde{\beta}(A)$$

for infinitely many n. So, we would have

$$\begin{aligned} & \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i) - f_*(X_i)| \\ & \quad \geq \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i) - f_*(X_i)|\, I(X_i \in A) \end{aligned}$$

$$
\begin{aligned}
&\geq \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A) \\
&\qquad \cdot \liminf_{n\to\infty} \frac{\sum_{i=1}^{n} |\hat{g}_n(X_i) - f_*(X_i)|\, I(X_i \in A)}{\max(1, \sum_{i=1}^{n} I(X_i \in A))} \\
&\geq \mathbb{P}(X_1 \in A)\tilde{\beta}(A) \\
&= \beta + \tilde{\beta} + 1. \qquad (4.15)
\end{aligned}
$$

On the other hand, we have

$$
\frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i) - f_*(X_i)| \leq \frac{1}{n}\sum_{i=1}^{n} |\hat{g}_n(X_i)| + \frac{1}{n}\sum_{i=1}^{n} |f_*(X_i)| \leq \beta + \tilde{\beta}
$$

for n sufficiently large, which contradicts (4.15). Thus, we must have $B \cap C^c = \emptyset$, proving Step 3.

Step 4 Let $e_0 = (0, 0, \ldots, 0)^T$ and e_i be the ith unit vector for $1 \leq i \leq d$. Let $v_* = (1/(4d), 1/d, \ldots, 1/d)$. Let A_i be defined as follows:

$$
\begin{aligned}
A_0 &= \left\{x \in [0,1]^d : \|x - e_0\| \leq \tau\right\}, \\
A_1 &= [1/2, 1] \times [0, 1] \times \cdots \times [0, 1] \subset [0, 1]^d, \\
A_i &= \left\{x \in [0,1]^d : \|x - e_i\| \leq \tau\right\} \text{ for } 2 \leq i \leq d, \\
A_{d+1} &= \left\{x \in [0,1]^d : \|x - v_*\| \leq \tau\right\}.
\end{aligned}
$$

We will show that there exists a positive constant τ such that for any y in A_{d+1} and x_i in A_i for $0 \leq i \leq d$, there exist nonnegative real numbers $p^0, p^1, \ldots, p^d$ summing to one such that

$$
p^0 x_0 + p^1 x_1 + \cdots + p^d x_d = y
$$

and that $p_1 \geq 1/(16d)$.

To fill in the details, let $y = (y^1, \ldots, y^d)$ be any point in A_{d+1} and $x_i = (x_i^1, \ldots, x_i^d)$ be any point in A_i for $0 \leq i \leq d$. We will show that there exists a nonnegative solution

$p^0, p^1, \ldots, p^d$ (summing to one) to the linear system

$$p^0 x_0 + p^1 x_1 + \cdots + p^d x_d = y$$

with $p^1 \geq 1/(16d)$.

Or equivalently, we will show that there exists a nonnegative solution $p^1, \ldots, p^d$ (summing less than or equal to one) to the linear system

$$\sum_{i=1}^{d} p^i (x_i - x_0) = y - x_0. \tag{4.16}$$

The linear system can be reexpressed as $Fp = y - x_0$, where $p = (p^1, \ldots, p^d)^T$ and $F = (F_{ij} : 1 \leq i, j \leq d)$ is a square $d \times d$ matrix in which the ith column is $x_i - x_0$ for $1 \leq i \leq d$. Note that F is invertible for sufficiently small $\tau > 0$ because we have

$$|F_{ii}| \, |F_{jj}| > \left(\sum_{k=1, k \neq i}^{n} F_{ik} \right) \left(\sum_{k=1, k \neq j}^{n} F_{jk} \right)$$

for all $i \neq j$ and $1 \leq i, j \leq d$ with sufficiently small τ, and hence, Theorem V of Taussky (1949) applies.

So, there exists a solution $p^1, \ldots, p^d$ to (4.16). To show that $p^1, \ldots, p^d$ are nonnegative, sum less than or equal to one, and $p^1 \geq 1/(16d)$, we let $G = (G_{ij} : 1 \leq i, j \leq d)$ be a square $d \times d$ matrix in which the first column is x_1 and the ith column is e_i for $2 \leq i \leq d$. Observe that $q^1, \ldots, q^d$ defined by

$$\begin{aligned} q^1 &= y^1 / x_1^1 \\ q^i &= y^i - y^1 x_1^i / x_1^1 \end{aligned}$$

for $2 \leq i \leq d$ satisfy $Gq = y$, where $q = (q^1, \ldots, q^d)$. Note also that $1/(8d) \leq q^1 \leq 2y^1$ and $1/(8d) \leq q^i \leq y^i$ for $2 \leq i \leq d$ for τ sufficiently small.

Set $|||F||| \triangleq \sup_{\|x\|=1} \|Fx\|$. Mapping a $d \times d$ square matrix to its inverse is continuous with respect to $|||\cdot|||$ in a neighborhood of F because F is invertible. Thus, we can make $|||F^{-1} - G^{-1}|||$ sufficiently small by making $|||F - G|||$ or τ sufficiently small. Also, $|||F^{-1}||| \leq 1/|||F||| \leq 1/\max_{1\leq i,j\leq d} |F_{ij}| \leq 4$ for τ sufficiently small. So,

$$\begin{aligned}
\|p - q\| &= \|F^{-1}(y - x_0) - G^{-1}y\| \\
&= \|(F^{-1} - G^{-1})y - F^{-1}x_0\| \\
&\leq \|(F^{-1} - G^{-1})y\| + \|F^{-1}x_0\| \\
&\leq |||F^{-1} - G^{-1}||| \cdot \|y\| + |||F^{-1}||| \cdot \|x_0\| \\
&\leq |||F^{-1} - G^{-1}||| + |||F^{-1}|||\tau,
\end{aligned}$$

and hence, $\|p-q\| \leq 1/(16d)$ for sufficiently small τ. Thus, $p^1, \ldots, p^d$ are nonnegative and sum less than or equal to one, and $p^1 \geq 1/(16d)$. Step 4 is proved.

Step 5 Let u_i be the vector identical to e_i except that its first element is one minus e_i's first element for $0 \leq i \leq d$. Let $w_* = (1 - 1/(4d), 1/d, \ldots, 1/d)$. Let B_i be defined as follows:

$$\begin{aligned}
B_0 &= \left\{x \in [0,1]^d : \|x - u_0\| \leq \tau\right\}, \\
B_1 &= [0, 1/2] \times [0,1] \times \cdots \times [0,1] \subset [0,1]^d, \\
B_i &= \left\{x \in [0,1]^d : \|x - u_i\| \leq \tau\right\} \text{ for } 2 \leq i \leq d, \\
B_{d+1} &= \left\{x \in [0,1]^d : \|x - w_*\| \leq \tau\right\}.
\end{aligned}$$

Then, there exists a positive constant τ such that for any y in B_{d+1} and x_i in B_i for $0 \leq i \leq d$, there exist nonnegative real numbers $p^0, p^1, \ldots, p^d$ summing to one such that

$$p^0 x_0 + p^1 x_1 + \cdots + p^d x_d = y$$

and $p^1 \geq 1/(16d)$. The proof of Step 5 is similar to the proof of Step 4 and is omitted.

Step 6 There exists a constant $\tilde{\gamma}$ such that

$$\inf_{x\in[0,1]^d, \hat{g}_n \in \mathcal{S}_n} \hat{g}_n(x) \geq \tilde{\gamma}$$

a.s. for n sufficiently large.

First, we show that

$$\inf_{x\in A_1, \hat{g}_n \in \mathcal{S}_n} \hat{g}_n(x) \geq \tilde{\gamma}$$

a.s. for n sufficiently large. Then it will follow similarly that

$$\inf_{x\in B_1, \hat{g}_n \in \mathcal{S}_n} \hat{g}_n(x) \geq \tilde{\gamma}$$

a.s. for n sufficiently large.

By Step 3, there exists a positive constant γ such that

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x\in A_i} |\hat{g}_n(x) - f_*(x)| \leq \gamma$$

a.s. for all $0 \leq i \leq d+1$ and n sufficiently large.

Since $|f_*(x)| \leq M$ for $x \in [0,1]^d$ by A5, we have

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x\in A_i} |\hat{g}_n(x)| \quad \leq \quad M + \gamma \tag{4.17}$$

a.s. for all $0 \leq i \leq d+1$ and n sufficiently large.

Set $\tilde{\gamma} = -32d(M + \gamma + 1)$. Note that for any $\hat{g}_n \in \mathcal{S}_n$, if $\hat{g}_n(x_1) \leq \tilde{\gamma}$ for some $x_1 \in A_1$ and $\hat{g}_n(x_i) \leq (\gamma + M + 1)$ for some $x_i \in A_i$ $(i = 0, 2, \ldots, d)$, then Step 4 guarantees that for any y in A_{d+1}, there exist nonnegative real numbers $p^0, p^1, \ldots, p^d$

summing to one that satisfy

$$p^0 x_0 + p^1 x_1 + \cdots + p^d x_d = y.$$

So, we have

$$\begin{aligned}
\hat{g}_n(y) &= \hat{g}_n(p^0 x_0 + \cdots + p^d x_d) \\
&\leq p^0 \hat{g}_n(x_0) + p^1 \hat{g}_n(x_1) + \cdots + p^d \hat{g}_n(x_d) \quad \text{because } \hat{g}_n \text{ is convex} \\
&\leq \tilde{\gamma}/(16d) + (M + \gamma + 1) \\
&= -(M + \gamma + 1).
\end{aligned}$$

So, if $\hat{g}_n(x) \leq \tilde{\gamma}$ for some $x \in A_1$, then we should either have

$$\inf_{x \in A_i} \hat{g}_n(x) \geq M + \gamma + 1$$

for some $i \in \{0, 2, \ldots, d\}$ or

$$\sup_{x \in A_{d+1}} \hat{g}_n(x) \leq -(M + \gamma + 1).$$

Thus,

$$\begin{aligned}
&\mathbb{P}\left(\inf_{x \in A_1, \hat{g}_n \in \mathcal{S}_n} \hat{g}_n(x) \leq \tilde{\gamma} \text{ for infinitely many } n \right) \\
&\leq \sum_{i=0,2,\ldots,d} \mathbb{P}\left(\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in A_i} \hat{g}_n(x) \geq M + \gamma + 1 \text{ for infinitely many } n \right) \\
&\quad + \mathbb{P}\left(\inf_{\hat{g}_n \in \mathcal{S}_n} \sup_{x \in A_{d+1}} \hat{g}_n(x) \leq -(M + \gamma + 1) \text{ for infinitely many } n \right) \\
&\leq \sum_{i=0,2,\ldots,d} \mathbb{P}\left(\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in A_i} |\hat{g}_n(x)| \geq M + \gamma + 1 \text{ for infinitely many } n \right) \\
&\quad + \mathbb{P}\left(\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in A_{d+1}} |\hat{g}_n(x)| \geq M + \gamma + 1 \text{ for infinitely many } n \right)
\end{aligned}$$

$$= \quad 0$$

by (4.17), proving Step 6.

Step 7 We prove that for any $c > 0$ there exists a positive constant $\tilde{\gamma}(c)$ such that

$$\sup_{x \in \mathcal{H}_c, \hat{g}_n \in \mathcal{S}_n} \hat{g}_n(x) \leq \tilde{\gamma}(c)$$

a.s. for n sufficiently large, where $\mathcal{H}_c = [c, 1-c]^d$.

First we prove that there exists a positive constant $\tau(c)$ such that for any $y \in \mathcal{H}_c$ and for any $x_i \in C_i$ $(1 \leq i \leq d)$, where

$$C_i \quad = \quad \left\{ x \in [0,1]^d : \|x - e_i\| \leq \tau(c) \right\},$$

there exist nonnegative real numbers $p^1, \ldots, p^d$ such that

$$p^1 x_1 + \cdots + p^d x_d = y$$

and that $p^i \leq 1$ for $1 \leq i \leq d$.

To fill in the details, note that we need to show that there exists a solution $p = (p^1, \ldots, p^d)^T$ to the linear equation

$$Hp \quad = \quad y \tag{4.18}$$

with $0 \leq p^i \leq 1$ for $1 \leq i \leq d$, where $H = (H_{ij} : 1 \leq i, j \leq d)$ is a square $d \times d$ matrix in which the ith column is x_i for $1 \leq i \leq d$. Set $\|H\|_\infty = \max_{1 \leq i \leq d} \sum_{j=1}^d |H_{ij}|$ and note that $\|H - I_d\|_\infty \leq \tau(c)$, where I_d is the $d \times d$ identity matrix. Hence, for $\tau(c) < 1/2$, H is invertible and we have

$$\|H^{-1}\|_\infty = \|(I_d + H - I_d)^{-1}\|_\infty \leq (1 - \|H - I_d\|_\infty)^{-1} \leq 2.$$

Therefore,

$$\|p - y\|_\infty = \|H^{-1}y - y\|_\infty \leq \|H^{-1} - I_d\|_\infty \|y\|_\infty \leq \|H^{-1} - I_d^{-1}\|_\infty.$$

Since mapping a $d \times d$ matrix to its inverse matrix is continuous with respect to $\|\cdot\|_\infty$ in a neighborhood of H and $\|H - I_d\|_\infty \leq \tau(c)$, there exists a positive number $\tau(c)$ that guarantees $\|H^{-1} - I_d^{-1}\| \leq c/2$. So, for such $\tau(c)$, $\|p - y\|_\infty \leq c/2$. Since $y \in [c, 1-c]^d$, $p^1, \ldots, p^d$ are nonnegative and each of them is less than or equal to one.

Now we prove Step 7. For $1 \leq i \leq d$, $r > 0$, and $\hat{g}_n \in \mathcal{S}_n$,

$$\begin{aligned} &\frac{1}{n}\sum_{j=1}^{n} I\left(X_j \in C_i, |\hat{g}_n(X_j)| \leq r\right) \\ &\quad \geq \quad \frac{1}{n}\sum_{j=1}^{n} I\left(X_j \in C_i\right) - \frac{1}{n}\sum_{j=1}^{n} I\left(X_j \in C_i, |\hat{g}_n(X_j)| > r\right). \end{aligned}$$

However, Markov inequality and Step 2 imply that

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \frac{1}{n}\sum_{j=1}^{n} I\left(X_j \in C_i, |\hat{g}_n(X_j)| > r\right) \quad \leq \quad \sup_{\overline{g_n} \in \mathcal{S}_n} r^{-1}\frac{1}{n}\sum_{j=1}^{n} |\hat{g}_n(X_j)| \leq \beta/r$$

a.s. for n sufficiently large. Choose r_0 so large that $\beta/r_0 \leq \overline{\gamma} \triangleq \min\{\mathbb{P}(X_1 \in C_i) : 1 \leq i \leq d\}/2$, then

$$\inf_{\hat{g}_n \in \mathcal{S}_n} \frac{1}{n}\sum_{j=1}^{n} I\left(X_j \in C_i, |\hat{g}_n(X_j)| \leq r_0\right) \geq \overline{\gamma}$$

a.s. for n sufficiently large.

For each such n, there exists $X_{I(i)} \in C_i$ with $1 \leq I(i) \leq n$ and $\left|\hat{g}_n(X_{I(i)})\right| \leq r_0$. For each $y \in [c, 1-c]^d$ and $X_{I(i)} \in C_i$ for $1 \leq i \leq d$, there exist $p^1, \ldots, p^d$ such that

$$y = p^1 X_{I(1)} + \cdots + p^d X_{I(d)}$$

and that $0 \leq p^i \leq 1$ for $1 \leq i \leq d$. So, the convexity of $\hat{g}_n$ yields

$$\hat{g}_n(y) \leq p^1 \hat{g}_n(X_{I(1)}) + \cdots + p^d \hat{g}_n(X_{I(d)}) \leq dr_0,$$

proving that

$$\sup_{x \in [c,1-c]^d, \hat{g}_n \mathcal{S}_n} \hat{g}_n(x) \leq dr_0$$

a.s. for n sufficiently large.

Step 8 Observe that the a.s. bound on $|\hat{g}_n|$ and $|f_*|$ uniformly in n over $\mathcal{H}_{c/2} = [c/2, 1-c/2]^d$ implies that $\hat{g}_n$ and f_* is Lipschitz over $\mathcal{H}_c = [c, 1-c]^d$ uniformly in n a.s. In particular, there exists a positive constant $\alpha(c)$ such that

$$\sup_{\hat{g}_n \in \mathcal{S}_n} |\hat{g}_n(x) - \hat{g}_n(y)| \leq \alpha(c)\|x - y\|$$

and

$$|f_*(x) - f_*(y)| \leq \alpha(c)\|x - y\|$$

for $x, y \in \mathcal{H}_c$ a.s. for n sufficiently large; see, for example, Roberts and Varberg (1974).

Step 9 Let

$$\begin{aligned} \mathcal{C}_c \;=\; & \{h : \mathcal{H}_c \to \mathbb{R} \text{ such that } h \text{ is convex on } \mathcal{H}_c, \\ & |h(x)| \leq |\tilde{\gamma}| + \tilde{\gamma}(c) \text{ and } |h(x) - h(y)| \leq \alpha(c)\|x - y\| \text{ for } x, y \in \mathcal{H}_c\}. \end{aligned}$$

Note that Steps 6, 7, and 8 guarantee that for each $c \geq 0$ there exists $n(c)$ such that $n \geq n(c)$ and $\hat{g}_n \in \mathcal{S}_n$ imply that $\hat{g}_n$ restricted to $\mathcal{H}_c$ belongs to $\mathcal{C}_c$ a.s. Furthermore, $\mathcal{C}_c$ is compact in the uniform metric d_c given by

$$d_c(h_1, h_2) = \sup_{x \in \mathcal{H}_c} |h_1(x) - h_2(x)|.$$

It follows that for each $\epsilon > 0$, there exists a finite collection of functions $h_1, \ldots, h_m$ in $\mathcal{C}_c$ such that

$$\bigcup_{i=1}^{m} \{h \in \mathcal{C}_c : d_c(h_i, h) < \epsilon\} \supseteq \mathcal{C}_c.$$

That is, $h_1, h_2, \ldots, h_m$ is an ϵ–net for $\mathcal{C}_c$; see Theorem 6 of Bronshtein (1976).

Step 10 We will prove that for any positive real numbers ϵ and δ and for any $z \in [0,1]^d$ and

$$B(z, \delta) \triangleq \left\{x \in [0,1]^d : \|x - z\| \leq \delta\right\},$$

we have

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in B(z,\delta)} (f_*(x) - \hat{g}_n(x)) \leq \epsilon$$

a.s. for n sufficiently large.

To fill in the details, let

$$C = \{ \sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in B(z,\delta)} (f_*(x) - \hat{g}_n(x)) \leq \epsilon \text{ for } n \text{ sufficiently large}\}.$$

We will prove that $\mathbb{P}(C) = 1$ by showing that $\mathbb{P}(A \cap B \cap C^c) = \emptyset$, where A and B are defined as follows and $\mathbb{P}(A) = \mathbb{P}(B) = 1$.

Let

$$A = \left\{ \frac{1}{n} \sum_{i=1}^{n} I(X_i \in B(z,\delta), -\epsilon/2 \leq \varepsilon_i \leq 0) \geq \eta/2 \text{ for } n \text{ sufficiently large} \right\},$$

where $\eta \triangleq \mathbb{P}(X_1 \in B(z,\delta), -\epsilon/2 \leq \varepsilon_1 \leq 0)$. By the strong law of large numbers, $\mathbb{P}(A) = 1$.

On the other hand, the dominated convergence theorem guarantees that

$$\mathbb{E}(I(X_1 \text{ is not in } \mathcal{H}_\delta)) \to 0$$

as $\delta \to 0$ because $I(X_1 \text{ is not in } \mathcal{H}_\delta) \downarrow 0$ a.s. as $\delta \downarrow 0$. So, take δ_0 small enough so

that

$$\mathbb{E}(I(X_1 \text{ is not in } \mathcal{H}_{\delta_0})) \;\leq\; \frac{\epsilon\eta}{24(M+|\tilde{\gamma}|)} \tag{4.19}$$

and note that

$$\frac{1}{n}\sum_{i=1}^{n} I(X_i \text{ is not in } \mathcal{H}_{\delta_0}) \leq \frac{\epsilon\eta}{12(M+|\tilde{\gamma}|)}$$

a.s. for n sufficiently large by the strong law of large numbers. Also, by Step 6 and A5, we have $(f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ \leq M + |\tilde{\gamma}|$ a.s. for n sufficiently large, so

$$\frac{1}{n}\sum_{i=1}^{n}(f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ I(X_i \text{ is not in } \mathcal{H}_{\delta_0}) \leq \epsilon\eta/12$$

a.s. for n sufficiently large.

Let $h_1, \ldots, h_m$ be an $\epsilon\eta/12$–net for $\mathcal{H}_{\delta_0}$. For each $j \in \{1, \ldots, m\}$, the strong law of large numbers guarantees that

$$\frac{1}{n}\sum_{i=1}^{n}(f_*(X_i) - h_j(X_i) - \epsilon/2)^+(1/2 - I(\varepsilon_i \leq 0))I(X_i \in \mathcal{H}_{\delta_0}) \to 0$$

as $n \to \infty$ because the X_is and the ε_is are independent and ε_i's have zero median. So,

$$\max_{1\leq j\leq m}\left|\frac{1}{n}\sum_{i=1}^{n}(f_*(X_i) - h_j(X_i) - \epsilon/2)^+(1/2 - I(\varepsilon_i \leq 0))I(X_i \in \mathcal{H}_{\delta_0})\right| \leq \epsilon\eta/24$$

a.s. for n sufficiently large.

We let

$$\begin{aligned} B \;=\; & \left\{\frac{1}{n}\sum_{i=1}^{n}(f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ I(X_i \text{ is not in } \mathcal{H}_{\delta_0}) \leq \epsilon\eta/12 \text{ for } n\right. \\ & \text{sufficiently large} \\ & \left.\max_{1\leq j\leq m}\left|\frac{1}{n}\sum_{i=1}^{n}(f_*(X_i) - h_j(X_i) - \epsilon/2)^+(1/2 - I(\varepsilon_i \leq 0))I(X_i \in \mathcal{H}_{\delta_0})\right|\right. \end{aligned}$$

$\leq \epsilon\eta/24$ for n sufficiently large$\}$

Since $\mathbb{P}(A) = \mathbb{P}(B) = 1$, it remains to show that $A \cap B \cap C^c = \emptyset$.

Suppose, on the contrary, that $\omega \in A \cap B \cap C^c$. Then for such ω, there exists $\hat{g}_n \in \mathcal{S}_n$ such that

$$\inf_{x \in B(z,\delta)} (f_*(x) - \hat{g}_n(x)) \quad > \quad \epsilon \tag{4.20}$$

for infinitely many n.

Define $k_n : [0,1]^d \to \mathbb{R}$ by $k_n = \max(f_*(x) - \epsilon/2, \hat{g}_n(x))$ for $x \in [0,1]^d$. Since k_n is convex, we must have

$$\varphi_n(k_n) \geq \varphi_n(\hat{g}_n),$$

or equivalently,

$$\begin{aligned} 0 &\leq \varphi_n(k_n) - \varphi(\hat{g}_n) \\ &= \frac{1}{n}\sum_{i=1}^{n} |Y_i - k_n(X_i)| - \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{g}_n(X_i)| \\ &= \frac{1}{n}\sum_{X_i \in P_n} |Y_i - k_n(X_i)| - \frac{1}{n}\sum_{X_i \in P_n} |Y_i - \hat{g}_n(X_i)|, \end{aligned}$$

where $P_n = \left\{x \in [0,1]^d : f_*(x) - \epsilon/2 \geq \hat{g}_n(x)\right\}$.

We denote

$$\begin{aligned} Q_{i,n} &= \{X_i \in P_n\} \cap \{\varepsilon_i + \epsilon/2 < -(f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)\} \\ R_{i,n} &= \{X_i \in P_n\} \cap \{-(f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2) \leq \varepsilon_i + \epsilon/2 < 0\} \\ S_{i,n} &= \{X_i \in P_n\} \cap \{0 \leq \varepsilon_i + \epsilon/2\} \end{aligned}$$

for $1 \leq i \leq n$ and observe that

$$
\begin{aligned}
0 &\leq \varphi_n(k_n) - \varphi(\hat{g}_n) \\
&= \frac{1}{n}\sum_{X_i \in P_n} |Y_i - (f_*(X_i) - \epsilon/2)| - \frac{1}{n}\sum_{X_i \in P_n} |Y_i - \hat{g}_n(X_i)| \\
&= \frac{1}{n}\sum_{X_i \in P_n} |\varepsilon_i + \epsilon/2| - \frac{1}{n}\sum_{X_i \in P_n} |\varepsilon_i + \epsilon/2 + (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)| \\
&= \frac{1}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)\, I(Q_{i,n}) \\
&\quad -\frac{1}{n}\sum_{i=1}^{n} (2\varepsilon_i + \epsilon + f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)\, I(R_{i,n}) \\
&\quad -\frac{1}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)\, I(S_{i,n}) \\
&= -\frac{1}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (1 - 2I(Q_{i,n})) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (\varepsilon_i + \epsilon/2)\, I(R_{i,n}) \\
&= -\frac{2}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (1/2 - I(Q_{i,n})) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (\varepsilon_i + \epsilon/2)\, I(R_{i,n}) \\
&= -\frac{2}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (I(\varepsilon_i \leq 0) - I(Q_{i,n})) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (1/2 - I(\varepsilon_i \leq 0)) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (\varepsilon_i + \epsilon/2)\, I(R_{i,n}) \\
&= -\frac{2}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (I(R_{i,n}) + I(-\epsilon/2 \leq \varepsilon_i \leq 0)) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) - \epsilon/2)^+ (1/2 - I(\varepsilon_i \leq 0)) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n} (\varepsilon_i + \epsilon/2)\, I(R_{i,n})
\end{aligned}
$$

$$
\begin{aligned}
&= -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\varepsilon_i\right)I(R_{i,n}) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ I(-\epsilon/2\le\varepsilon_i\le 0) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0)) \\
&\le -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ I(-\epsilon/2\le\varepsilon_i\le 0) \\
&\quad -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0)) = \mathrm{I}+\mathrm{II},\ \text{say.} \quad (4.21)
\end{aligned}
$$

By (4.20), we have

$$
\begin{aligned}
\mathrm{I} &= -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ I(-\epsilon/2\le\varepsilon_i\le 0) \\
&\le -\frac{1}{n}\sum_{i=1}^{n}\epsilon I(X_i\in B(z,\delta),-\epsilon/2\le\varepsilon_i\le 0)
\end{aligned}
$$

for infinitely many n.

Since $\omega\in A$,

$$
\mathrm{I} = -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ I(-\epsilon/2\le\varepsilon_i\le 0)\le -\epsilon\eta/2 \quad (4.22)
$$

for infinitely many n.

On the other hand, note that for each $1\le j\le m$,

$$
\begin{aligned}
\mathrm{II} &= -(2/n)\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0)) \\
&= -(2/n)\sum_{X_i \text{ is not in } \mathcal{H}_{\delta_0}}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0)) \\
&\quad -(2/n)\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0))I(X_i\in\mathcal{H}_{\delta_0}) \\
&\le -(2/n)\sum_{X_i \text{ is not in } \mathcal{H}_{\delta_0}}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+ (1/2-I(\varepsilon_i\le 0))
\end{aligned}
$$

$$
\begin{aligned}
&-(2/n)\sum_{i=1}^{n}\left(f_*(X_i)-h_j(X_i)-\epsilon/2\right)^+\left(1/2-I(\varepsilon_i\le 0)\right)I(X_i\in\mathcal{H}_{\delta_0})\\
&+(2/n)\sum_{i=1}^{n}\left|h_j(X_i)-\hat{g}_n(X_i)\right|\left|1/2-I(\varepsilon_i\le 0)\right|I(X_i\in\mathcal{H}_{\delta_0})\\
&\text{because } -(a+b)^+c\le -a^+c+|b||c| \text{ for } a,b,c\in\mathbb{R}\\
\le\ & (2/n)\sum_{X_i \text{ is not in } \mathcal{H}_{\delta_0}}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+\left|1/2-I(\varepsilon_i\le 0)\right|\\
&+2\max_{1\le j\le m}\left|\frac{1}{n}\sum_{X_i\in\mathcal{H}_{\delta_0}}\left(f_*(X_i)-h_j(X_i)-\epsilon/2\right)^+\left(1/2-I(\varepsilon_i\le 0)\right)\right|\\
&+(2/n)\sum_{i=1}^{n}\sup_{x\in\mathcal{H}_{\delta_0}}\left|h_j(x)-\hat{g}_n(x)\right|\left|1/2-I(\varepsilon_i\le 0)\right|I(X_i\in\mathcal{H}_{\delta_0}). \qquad (4.23)
\end{aligned}
$$

Since (4.23) holds for any $j\in\{1,\ldots,m\}$,

$$
\begin{aligned}
\text{II} \ &\le\ (2/n)\sum_{X_i \text{ is not in } \mathcal{H}_{\delta_0}}\left(f_*(X_i)-\hat{g}_n(X_i)-\epsilon/2\right)^+\left|1/2-I(\varepsilon_i\le 0)\right|\\
&+\ 2\max_{1\le j\le m}\left|\frac{1}{n}\sum_{X_i\in\mathcal{H}_{\delta_0}}\left(f_*(X_i)-h_j(X_i)-\epsilon/2\right)^+\left(1/2-I(\varepsilon_i\le 0)\right)\right|\\
&+\ \epsilon\eta/12\\
&\le\ \epsilon\eta/12+\epsilon\eta/12+\epsilon\eta/12 \quad \text{because } \omega\in B\\
&=\ \epsilon\eta/4 \qquad (4.24)
\end{aligned}
$$

a.s. for n sufficiently large.

Combination of (4.21), (4.22), and (4.24) gives $0\le\varphi(k_n)-\varphi(\hat{g}_n)\le-\epsilon\eta/2$ for infinitely many n, which is a contradiction. This proves that $A\cap B\cap C^c=\emptyset$ and that $\mathbb{P}(C)=1$.

Step 11 We will prove that for any positive real numbers ϵ and δ and for any $z\in[0,1]^d$ and

$$B(z,\delta)\triangleq\left\{x\in[0,1]^d:\|x-z\|\le\delta\right\},$$

we have

$$\sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in B(z,\delta)} (\hat{g}_n(x) - f_*(x)) \leq \epsilon$$

a.s. for n sufficiently large.

To fill in the details, let

$$C = \{ \sup_{\hat{g}_n \in \mathcal{S}_n} \inf_{x \in B(z,\delta)} (\hat{g}_n(x) - f_*(x)) \leq \epsilon \text{ for } n \text{ sufficiently large} \}.$$

We will prove that $\mathbb{P}(C) = 1$ by showing that $\mathbb{P}(A \cap B \cap C^c) = \emptyset$, where A and B are defined as follows and $\mathbb{P}(A) = \mathbb{P}(B) = 1$.

Let

$$A = \left\{ \frac{1}{n} \sum_{i=1}^{n} I(X_i \in B(z,\delta), 0 < \varepsilon_i < \epsilon/2) \geq \eta/2 \text{ for } n \text{ sufficiently large} \right\},$$

where $\eta \triangleq \mathbb{P}(X_1 \in B(z,\delta), 0 < \varepsilon_1 < \epsilon/2)$. By the strong law of large numbers, $\mathbb{P}(A) = 1$.

On the other hand, the strong law of large numbers and A4 ensure that

$$\frac{1}{n} \sum_{i=1}^{n} (1/2 - I(\varepsilon_i > 0)) = \frac{1}{n} \sum_{i=1}^{n} (I(\varepsilon_i \leq 0) - 1/2) \geq -\eta/16$$

a.s. for n sufficiently large. Also, similar arguments leading to (4.24) ensure that

$$\frac{1}{n} \sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) + \epsilon)^+ (1/2 - I(\varepsilon \leq 0)) \geq -\epsilon\eta/16$$

a.s. as $n \to \infty$.

So, if we let

$$B = \left\{ \frac{1}{n} \sum_{i=1}^{n} (1/2 - I(\varepsilon_i > 0)) \geq -\eta/16 \text{ for } n \text{ sufficiently large} \right.$$

$$\frac{1}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)^+\left(1/2-I(\varepsilon<0)\right)\geq -\epsilon\eta/16$$
$$\text{for } n \text{ sufficiently large}\Big\},$$

then $\mathbb{P}(B)=1$.

Since $\mathbb{P}(A)=\mathbb{P}(B)=1$, it remains to show that $A\cap B\cap C^c=\emptyset$.

Suppose, on the contrary, that $\omega\in A\cap B\cap C^c$. Then for such ω, there exists $\hat{g}_n\in\mathcal{S}_n$ such that

$$\inf_{x\in B(z,\delta)}\left(\hat{g}_n(x)-f_*(x)\right) \;>\; \epsilon \tag{4.25}$$

for infinitely many n.

Define $k_n:[0,1]^d\to\mathbb{R}$ by $k_n(x)=\max(\hat{g}_n(x)-\epsilon, f_*(x))$ for $x\in[0,1]^d$. Since k_n is convex, we must have

$$\varphi_n(k_n)\geq\varphi_n(\hat{g}_n),$$

or equivalently,

$$\begin{aligned}
0 \;\leq\; & \varphi_n(k_n)-\varphi_n(\hat{g}_n)=\frac{1}{n}\sum_{i=1}^{n}|Y_i-k_n(X_i)|-\frac{1}{n}\sum_{i=1}^{n}|Y_i-\hat{g}_n(X_i)| \\
= \;& \frac{1}{n}\sum_{X_i\in P_n}|Y_i-\hat{g}_n(X_i)+\epsilon|-\frac{1}{n}\sum_{X_i\in P_n}|Y_i-\hat{g}_n(X_i)| \\
& +\frac{1}{n}\sum_{X_i\in P_n^c}|\varepsilon_i|-\frac{1}{n}\sum_{X_i\in P_n^c}|\varepsilon_i+f_*(X_i)-\hat{g}_n(X_i)| \\
= \;& \mathrm{I}+\mathrm{II}+\mathrm{III}+\mathrm{IV}, \text{say},
\end{aligned} \tag{4.26}$$

where $P_n=\left\{x\in[0,1]^d:\hat{g}_n(x)-\epsilon\geq f_*(x)\right\}$.

We denote

$$\begin{aligned}
Q_{i,n} &= \{X_i\in P_n\}\cap\{\varepsilon_i\geq-(f_*(X_i)-\hat{g}_n(X_i))\} \\
R_{i,n} &= \{X_i\in P_n\}
\end{aligned}$$

$$\cap \{-(f_*(X_i) - \hat{g}_n(X_i) + \epsilon) \leq \varepsilon_i < -(f_*(X_i) - \hat{g}_n(X_i))\}$$

$$S_{i,n} = \{X_i \in P_n\} \cap \{\varepsilon_i < -(f_*(X_i) - \hat{g}_n(X_i) + \epsilon)\}$$

and observe that

$$\begin{aligned}
\mathrm{I} + \mathrm{II} &= \frac{1}{n} \sum_{X_i \in P_n} |Y_i - \hat{g}_n(X_i) + \epsilon| - \frac{1}{n} \sum_{X_i \in P_n} |Y_i - \hat{g}_n(X_i)| \\
&= \frac{1}{n} \sum_{X_i \in P_n} |f_*(X_i) - \hat{g}_n(X_i) + \varepsilon_i + \epsilon| \\
&\quad - \frac{1}{n} \sum_{X_i \in P_n} |f_*(X_i) - \hat{g}_n(X_i) + \varepsilon_i| \\
&= \frac{1}{n} \sum_{i=1}^{n} \epsilon I(Q_{i,n}) + \frac{1}{n} \sum_{i=1}^{n} (2f_*(X_i) - 2\hat{g}_n(X_i) + 2\varepsilon_i + \epsilon) I(R_{i,n}) \\
&\quad - \frac{1}{n} \sum_{i=1}^{n} \epsilon I(S_{i,n}) \\
&= -\frac{1}{n} \sum_{X_i \in P_n} \epsilon \left(1 - 2I(S_{i,n}^c)\right) \\
&\quad + \frac{2}{n} \sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) + \varepsilon_i) I(R_{i,n}) \\
&= -\frac{2}{n} \sum_{X_i \in P_n} \epsilon \left(1/2 - I(S_{i,n}^c)\right) \\
&\quad + \frac{2}{n} \sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) + \varepsilon_i) I(R_{i,n}) \\
&= -\frac{2}{n} \sum_{X_i \in P_n} \epsilon \left(I(\varepsilon_i > 0) - I(S_{i,n}^c)\right) \\
&\quad - \frac{2}{n} \sum_{X_i \in P_n} \epsilon (1/2 - I(\varepsilon_i > 0)) \\
&\quad + \frac{2}{n} \sum_{i=1}^{n} (f_*(X_i) - \hat{g}_n(X_i) + \varepsilon_i) I(R_{i,n}) \\
&= -\frac{2}{n} \sum_{X_i \in P_n} \epsilon I\left(0 < \varepsilon_i < -(f_*(X_i) - \hat{g}_n(X_i) + \epsilon)\right) \\
&\quad - \frac{2}{n} \sum_{X_i \in P_n} \epsilon (1/2 - I(\varepsilon_i > 0))
\end{aligned}$$

$$
\begin{aligned}
&+\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\varepsilon_i\right)I(R_{i,n})\\
\leq\ & -\frac{2}{n}\sum_{X_i\in P_n}\epsilon I\left(0<\varepsilon_i<-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\right)\\
&-\frac{2}{n}\sum_{X_i\in P_n}\epsilon\left(1/2-I(\varepsilon_i>0)\right)\\
&+\frac{2}{n}\sum_{X_i\in P_n}\left(f_*(X_i)-\hat{g}_n(X_i)+\varepsilon_i\right)\\
&\cdot I(-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\leq\varepsilon_i<-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon/2\right))\\
\leq\ & -\frac{2}{n}\sum_{X_i\in P_n}\epsilon I\left(0<\varepsilon_i<-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\right)\\
&-\frac{2}{n}\sum_{X_i\in P_n}\epsilon\left(1/2-I(\varepsilon_i>0)\right)\\
&-\frac{1}{n}\sum_{X_i\in P_n}\epsilon I(-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\leq\varepsilon_i\\
&\qquad<-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon/2\right))\\
\leq\ & -\frac{1}{n}\sum_{X_i\in P_n}\epsilon I\left(0<\varepsilon_i<-\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon/2\right)\right)\\
&-\frac{2}{n}\sum_{X_i\in P_n}\epsilon\left(1/2-I(\varepsilon_i>0)\right)\\
\leq\ & -\frac{1}{n}\sum_{X_i\in P_n}\epsilon I\left(0<\varepsilon_i<\epsilon/2\right)-\frac{2}{n}\sum_{X_i\in P_n}\epsilon\left(1/2-I(\varepsilon_i>0)\right). \qquad (4.27)
\end{aligned}
$$

On the other hand, we have

$$
\begin{aligned}
\mathrm{III}+\mathrm{IV}&=\frac{1}{n}\sum_{X_i\in P_n^c}|\varepsilon_i|-\frac{1}{n}\sum_{X_i\in P_n^c}|\varepsilon_i+f_*(X_i)-\hat{g}_n(X_i)|\\
&=-\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)+\varepsilon_i\right)\\
&\cdot I\left(0\leq f_*(X_i)-\hat{g}_n(X_i),-\left(f_*(X_i)-\hat{g}_n(X_i)\right)<\varepsilon_i\leq 0\right)\\
&-\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)\right)\\
&\cdot\left(\frac{1}{2}I(0\leq f_*(X_i)-\hat{g}_n(X_i))-I\left(0\leq f_*(X_i)-\hat{g}_n(X_i),\ \varepsilon_i\leq 0\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)+\varepsilon_i\right) I\Big(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0,\\
&\qquad 0<\varepsilon_i\le -\left(f_*(X_i)-\hat{g}_n(X_i)\right)\Big)\\
&+\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)\right)\Big(\frac{1}{2}I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0\right)\\
&\qquad -I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0,\ \varepsilon_i>0\right)\Big).
\end{aligned}
$$

Since the first and the third terms in the above equations are always negative, we have

$$
\begin{aligned}
&\mathrm{III}+\mathrm{IV}\\
\le\ & -\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)\right)\Big(\frac{1}{2}I(0\le f_*(X_i)\\
&-\hat{g}_n(X_i))-I\left(0\le f_*(X_i)-\hat{g}_n(X_i),\ \varepsilon_i\le 0\right)\Big)\\
&+\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)\right)\Big(\frac{1}{2}I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0\right)\\
&\qquad -I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0,\ \varepsilon_i>0\right)\Big).\\
=\ & -\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\Big(\frac{1}{2}I(0\le f_*(X_i)-\hat{g}_n(X_i))\\
&\qquad -I\left(0\le f_*(X_i)-\hat{g}_n(X_i),\ \varepsilon_i\le 0\right)\Big)\\
&+\frac{2}{n}\sum_{X_i\in P_n^c}\epsilon\Big(\frac{1}{2}I(0\le f_*(X_i)-\hat{g}_n(X_i))\\
&-I\left(0\le f_*(X_i)-\hat{g}_n(X_i),\ \varepsilon_i\le 0\right)\Big)\\
&+\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\Big(\frac{1}{2}I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0\right)\\
&\qquad -I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0,\ \varepsilon_i>0\right)\Big).\\
&-\frac{2}{n}\sum_{X_i\in P_n^c}\epsilon\Big(\frac{1}{2}I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0\right)\\
&-I\left(-\epsilon<f_*(X_i)-\hat{g}_n(X_i)<0,\ \varepsilon_i>0\right)\Big).\\
=\ & -\frac{2}{n}\sum_{X_i\in P_n^c}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)\Big(\frac{1}{2}I(0\le f_*(X_i)-\hat{g}_n(X_i))
\end{aligned}
$$

$$
\begin{aligned}
&\quad -I\left(0 \le f_*(X_i) - \hat{g}_n(X_i),\ \varepsilon_i \le 0\right)\Big) \\
&-\frac{2}{n}\sum_{X_i \in P_n^c} \epsilon\Big(\frac{1}{2} I(0 \le f_*(X_i) - \hat{g}_n(X_i)) \\
&-I\left(0 \le f_*(X_i) - \hat{g}_n(X_i),\ \varepsilon_i > 0\right)\Big) \\
&-\frac{2}{n}\sum_{X_i \in P_n^c} \left(f_*(X_i) - \hat{g}_n(X_i) + \epsilon\right)\Big(\frac{1}{2} I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i) < 0\right) \\
&\quad -I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i) < 0,\ \varepsilon_i \le 0\right)\Big). \\
&-\frac{2}{n}\sum_{X_i \in P_n^c} \epsilon\Big(\frac{1}{2} I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i) < 0\right) \\
&\quad -I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i) < 0,\ \varepsilon_i > 0\right)\Big).
\end{aligned}
$$

Combining the first and third terms in the above expression and combining the second and fourth terms in the above expression yield

$$
\begin{aligned}
&\text{III} + \text{IV} \\
&\le \quad -\frac{2}{n}\sum_{X_i \in P_n^c} \left(f_*(X_i) - \hat{g}_n(X_i) + \epsilon\right)\Big(\frac{1}{2} I(-\epsilon < f_*(X_i) - \hat{g}_n(X_i)) \\
&\qquad\quad -I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i),\ \varepsilon_i \le 0\right)\Big) \\
&\qquad -\frac{2}{n}\sum_{X_i \in P_n^c} \epsilon\Big(\frac{1}{2} I(-\epsilon < f_*(X_i) - \hat{g}_n(X_i)) \\
&\qquad\quad -I\left(-\epsilon < f_*(X_i) - \hat{g}_n(X_i),\ \varepsilon_i > 0\right)\Big) \\
&= \quad -\frac{2}{n}\sum_{i=1}^{n} \left(f_*(X_i) - \hat{g}_n(X_i) + \epsilon\right)^+ \left(1/2 - I(\varepsilon_i \le 0)\right) \\
&\qquad -\frac{2}{n}\sum_{X_i \in P_n^c} \epsilon(1/2 - I(\varepsilon_i > 0)).
\end{aligned} \tag{4.28}
$$

From (4.27) and (4.28), we obtain

$$
\begin{aligned}
&\text{I} + \text{II} + \text{III} + \text{IV} \\
&\le \quad -\frac{1}{n}\sum_{X_i \in P_n} \epsilon I\left(0 < \varepsilon_i < \epsilon/2\right) - \frac{2}{n}\sum_{X_i \in P_n} \epsilon\left(1/2 - I(\varepsilon_i > 0)\right)
\end{aligned}
$$

$$
\begin{aligned}
& -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)^+\left(1/2-I(\varepsilon_i\le 0)\right) \\
& -\frac{2}{n}\sum_{X_i\in P_n^c}\epsilon(1/2-I(\varepsilon_i>0)) \\
= \quad & -\frac{1}{n}\sum_{i=1}^{n}\epsilon I\left(X_i\in P_n, 0<\varepsilon_i<\epsilon/2\right)-\frac{2}{n}\sum_{i=1}^{n}\epsilon\left(1/2-I(\varepsilon_i>0)\right) \\
& -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)^+\left(1/2-I(\varepsilon_i\le 0)\right). \qquad (4.29)
\end{aligned}
$$

By (4.25),

$$
-\frac{1}{n}\sum_{i=1}^{n}\epsilon I\left(X_i\in P_n, 0<\varepsilon_i<\epsilon/2\right)\le -\frac{1}{n}\sum_{i=1}^{n}\epsilon I\left(X_i\in B(z,\delta), 0<\varepsilon_i<\epsilon/2\right)
$$

for infinitely many n and because $\omega\in A$,

$$
-\frac{1}{n}\sum_{i=1}^{n}\epsilon I\left(X_i\in P_n, 0<\varepsilon_i<\epsilon/2\right) \quad\le\quad -\epsilon\eta/2 \qquad (4.30)
$$

for infinitely many n. However, because $\omega\in B$,

$$
\begin{aligned}
& -\frac{2}{n}\sum_{i=1}^{n}\epsilon\left(1/2-I(\varepsilon_i>0)\right) \\
& -\frac{2}{n}\sum_{i=1}^{n}\left(f_*(X_i)-\hat{g}_n(X_i)+\epsilon\right)^+\left(1/2-I(\varepsilon_i\le 0)\right) \\
& \le \epsilon\eta/4 \qquad (4.31)
\end{aligned}
$$

for n sufficiently large. Combination of (4.26), (4.29), (4.30), and (4.31) gives $0\le \varphi(k_n)-\varphi(\hat{g}_n)=\text{I+II+III+IV}\le -\epsilon\eta/4$ for infinitely many n, which is a contradiction. This proves that $A\cap B\cap C^c=\emptyset$ and that $\mathbb{P}(C)=1$.

Step 12 We will prove that for any $\epsilon>0$,

$$
\sup_{x\in\mathcal{H}_c,\hat{g}_n\in\mathcal{S}_n}\left(f_*(x)-\hat{g}_n(x)\right)\le\epsilon
$$

a.s. for n sufficiently large.

Take $\delta = \epsilon/(6\alpha(c))$, where $\alpha(c)$ is given as in Step 8. Since $\mathcal{H}_c$ is compact, there exist a finite number of points $y_1, \ldots, y_l$ in $\mathcal{H}_c$ such that $B_c(y_i, \delta) \triangleq \{x \in \mathcal{H}_c : \|x - y_i\| \leq \delta\}$ for $1 \leq i \leq l$ covers $\mathcal{H}_c$.

If there exists $\hat{g}_n \in \mathcal{S}_n$ such that $\sup_{x \in \mathcal{H}_c} (f_*(x) - \hat{g}_n(x)) > \epsilon$ for infinitely many n, for each of such n, there exists a point x_n in $\mathcal{H}_c$ such that

$$f_*(x_n) - \hat{g}_n(x_n) \quad > \quad \epsilon/2. \tag{4.32}$$

In this case, infinitely many of the x_ns will be in $B_c(y_j, \delta)$ for some j, so if we choose a subsequence $(n_k : k \geq 1)$ so that x_{n_k} is in $B_c(y_j, \delta)$ for all $k \geq 1$, then for any $x \in B_c(y_j, \delta)$ we have

$$\begin{aligned} & f_*(x) - \hat{g}_n(x) \\ &= f_*(x) - f_*(x_{n_k}) + f_*(x_{n_k}) - \hat{g}_n(x_{n_k}) + \hat{g}_n(x_{n_k}) - \hat{g}_n(x) \\ &\geq -\epsilon/6 + \epsilon/2 - \epsilon/6 \geq \epsilon/6 \end{aligned}$$

by (4.32).

So, $\sup_{x \in \mathcal{H}_c}(f_*(x) - \hat{g}_n(x)) > \epsilon$ implies $\inf_{x \in B(x_j, \delta)}(f_*(x) - \hat{g}_n(x)) \geq \epsilon/6$ for some j, and hence,

$$\begin{aligned} & \mathbb{P}\left(\sup_{x \in \mathcal{H}_c, \hat{g}_n \in \mathcal{S}_n} (f_*(x) - \hat{g}_n(x)) > \epsilon \text{ for infinitely many } n \right) \\ &= \sum_{j=1}^{l} \mathbb{P}\left(\sup_{\hat{g}_n \mathcal{S}_n} \inf_{x \in B(x_j, \delta)} (f_*(x) - \hat{g}_n(x)) \geq \epsilon/6 \text{ for infinitely many } n \right) \\ &= 0 \end{aligned}$$

by Step 10, proving Step 12.

Step 13 For any $\epsilon > 0$,

$$\sup_{x \in \mathcal{H}_c, \hat{g}_n \in \mathcal{S}_n} (\hat{g}_n(x) - f_*(x)) \leq \epsilon$$

a.s. for n sufficiently large.

The proof is similar to the proof of Step 12 (Step 11 is used instead of Step 10) and is omitted.

Step 14 Theorem 3 follows from Steps 12 and 13.

4.6.2 Proof of Theorem 4

It suffices to prove the second part of Theorem 4. The first part of Theorem 4 can be justified similarly to the second part. Suppose that f_* is differentiable on $[c, 1-c]^d$. Take $c_0 < c$ and let

$$A = \left\{ \sup_{x \in [c_0, 1-c_0]^d, \hat{g}_n \in \mathcal{S}_n} |\hat{g}_n(x) - f_*(x)| \to 0 \text{ as } n \to \infty \right\},$$

then $\mathbb{P}(A) = 1$ by Theorem 3. We will show that $\mathbb{P}(B) = 1$, where

$$B = \left\{ \sup_{x \in [c, 1-c]^d, \xi \in \partial \hat{g}_n(x), \hat{g}_n \in \mathcal{S}_n} \|\xi - \nabla f_*(x)\| \to 0 \text{ as } n \to \infty \right\},$$

by proving that $A \cap B^c = \emptyset$.

Suppose, on the contrary, that $\omega \in A \cap B^c$ exists. For such an ω, there exists $\epsilon > 0$, $x_n \in [c, 1-c]^d$, $\hat{g}_n \in \mathcal{S}_n$ and $\xi_n \in \partial \hat{g}_n(x_n)$ such that

$$\|\xi_n - \nabla f_*(x_n)\| > \epsilon$$

for infinitely many n. Furthermore, there exists an index $i \in \{1, \ldots, d\}$ such that

$$\left| e_i^T \xi_n - e_i^T \nabla f_*(x_n) \right| \;>\; \epsilon/d \tag{4.33}$$

for infinitely many n, where e_i is the ith unit vector. Equation (4.33) implies that either

$$e_i^T \xi_n \;>\; e_i^T \nabla f_*(x_n) + \epsilon/d \tag{4.34}$$

or

$$e_i^T \xi_n \;<\; e_i^T \nabla f_*(x_n) - \epsilon/d \tag{4.35}$$

holds. We first consider the case where (4.34) holds. Since $[c, 1-c]^d$ is compact, there exists a subsequence $(x_{n_k} : 1 \leq k)$ that converges to a point x_0 in $[c, 1-c]^d$. Passing to subsequences if necessary, for any $\lambda > 0$ small enough that $x_0 + \lambda e_i \in [(c+c_0)/2, 1-(c+c_0)/2]^d$, we have $x_n + \lambda e_i \in [c_0, 1-c_0]^d$ for all sufficiently large n and

$$e_i^T \xi_n \;\leq\; \left(\hat{g}_n(x_n + \lambda e_i) - \hat{g}_n(x_n)\right)/\lambda. \tag{4.36}$$

Since $\omega \in A$ and the $\hat{g}_n$s are continuous on $[c_0, 1-c_0]^d$, $\hat{g}_n(x_n + \lambda e_i)$ tends to $f_*(x_0 + \lambda e_i)$ and $\hat{g}_n(x_n)$ tends to $f_*(x_0)$ as $n \to \infty$. By Theorem 25.5 in p. 246 of Rockafellar (1970), ∇f_* is continuous on $[c, 1-c]^d$, and hence, $\nabla f_*(x_n)$ tends to $\nabla f_*(x_0)$. Therefore,

$$\begin{aligned} e_i^T \nabla f_*(x_0) + \epsilon/d \;&=\; \lim_{n\to\infty} e_i^T \nabla f_*(x_n) + \epsilon/d \\ &\leq\; \limsup_{n\to\infty} e_i^T \xi_n \text{ by (4.34)} \end{aligned}$$

$$\begin{aligned} &\leq \lim_{n\to\infty} \left(\hat{g}_n(x_n + \lambda e_i) - \hat{g}_n(x_n)\right)/\lambda \text{ by (4.36)} \\ &= \left(f_*(x_0 + \lambda e_i) - f_*(x_0)\right)/\lambda. \end{aligned} \tag{4.37}$$

This is supposed to hold for every sufficiently small $\lambda > 0$. But

$$e_i^T \nabla f_*(x_0) = \lim_{\lambda \downarrow 0} \left(f_*(x_0 + \lambda e_i) - f_*(x_0)\right)/\lambda,$$

which contradicts (4.37). Similar arguments can be applied to reach a contradiction in case of (4.35). Hence, Theorem 4 is proved.

Chapter 5

A Statistical Technique for the Initial Transient Problem

5.1. Overview

As we face increasingly complex systems in manufacturing, the service industry, and telecommunications, discrete-event simulations have become an essential tool for analyzing and evaluating them. Frequently, we wish to compute the steady-state performance of a system via discrete event simulations. While pursuing more efficient computation, standard estimators for steady-state performance invariably involve a bias that is induced by the initial transient phase at the beginning of simulation outputs. Before we proceed further, we will introduce three examples that require efficient computations of steady-state performance measures.

5.1.1 Motivation Examples

Example 1. Single server queueing systems A single server queueing system is one of the most commonly used queueing models. It captures the dynamics of a system with one server that serves incoming customers who wait in queue when not served immediately upon arrival. Quantities of interest such as the long-run average number of customers in the system or the long-run average amount of time a customer spends in the queue cannot be computed exactly even in the simplest setting of independent

interarrival and service times. Hence, simulation becomes a useful means to compute steady-state performance measures. When starting a simulation, we need to specify the initial number of customers in the system, the remaining service time for the customer in service, if any, and the remaining interarrival time. It is desirable to set these quantities according to the corresponding steady-state distributions, but the steady-state distributions are not known a priori and hence they are initialized in some arbitrary fashion at the simulator's convenience.

Suppose that we wish to compute the long-run average waiting time of customers in an M/M/1 queue which is equal to $\frac{\rho}{\mu(1-\rho)}$, where ρ is the traffic density defined by $\frac{\lambda}{\mu}$, λ is the arrival rate and μ is the service rate. Given $\lambda = 0.96$, $\mu = 1$, $\rho = 0.96$, the theoretical value of average waiting time in an M/M/1 queue should be 24. However, the system is initialized empty and idle. Then the average waiting time in the queue will increase as the system evolves over time until it reaches its steady-state, 24, after which it stays constant around 24; for details about transient behavior of M/M/1 queue, see Kelton and Law (1985). Figure 5.1 shows a trajectory of the waiting time of the nth customer in an M/M/1 queue with the arrival rate of 0.96 and the service rate of 1.

As depicted in Figure 5.1, a typical simulation output includes a transient phase at the beginning, inducing a significant bias when the long-run average waiting time in the queue is computed through the arithmetic mean of the observations in the simulation output. To fix this problem, our proposed algorithm will eliminate the observations until the steady-state behavior becomes apparent. Then we compute the arithmetic mean of the rest of the observations as an estimate of the steady-state mean. The truncation point after which the observations are retained for analysis will be chosen through simple and powerful proposed technique.

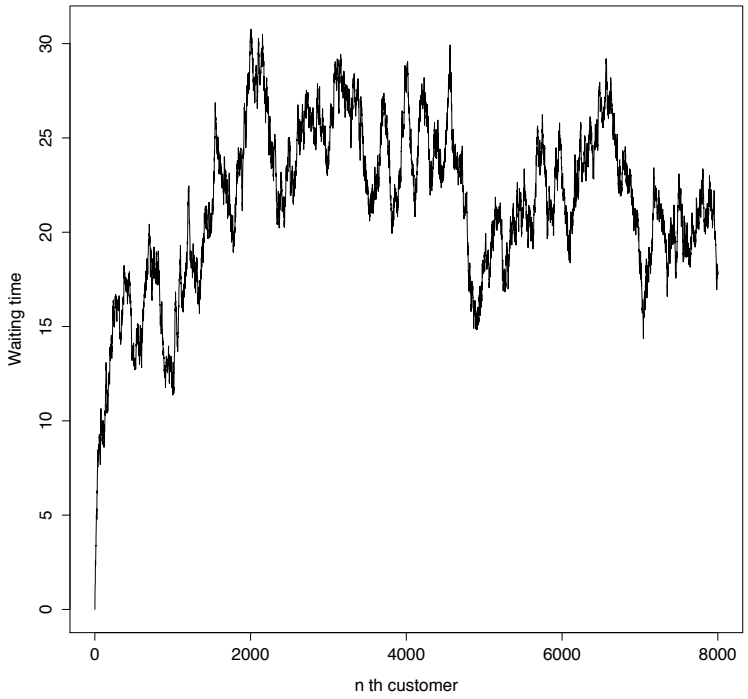

Figure 5.1: A trajectory of the waiting time of the nth customer in an M/M/1 queue

Example 2. Multi-station queueing networks In a multi-station queueing network with k stations, the long-run average number of customers in each station is usually computed by utilizing the simulated number of customers in each station over time $(X_1(t_n), \ldots, X_k(t_n) : n \geq 0)$, where t_n is the time when the nth customer departure occurs. In order to commence the simulation, the simulator needs to initialize the number of customers in each station, the remaining service times of customers in the stations, if any, and the remaining interarrival times of externally arriving customers, if any. With no information on steady-state distribution, these quantities are initialized arbitrarily, resulting in a transient phase in simulation output. Figure 5.2 shows a trajectory of the number of customers in each station when the nth customer departure occurs in a three-station queueing network.

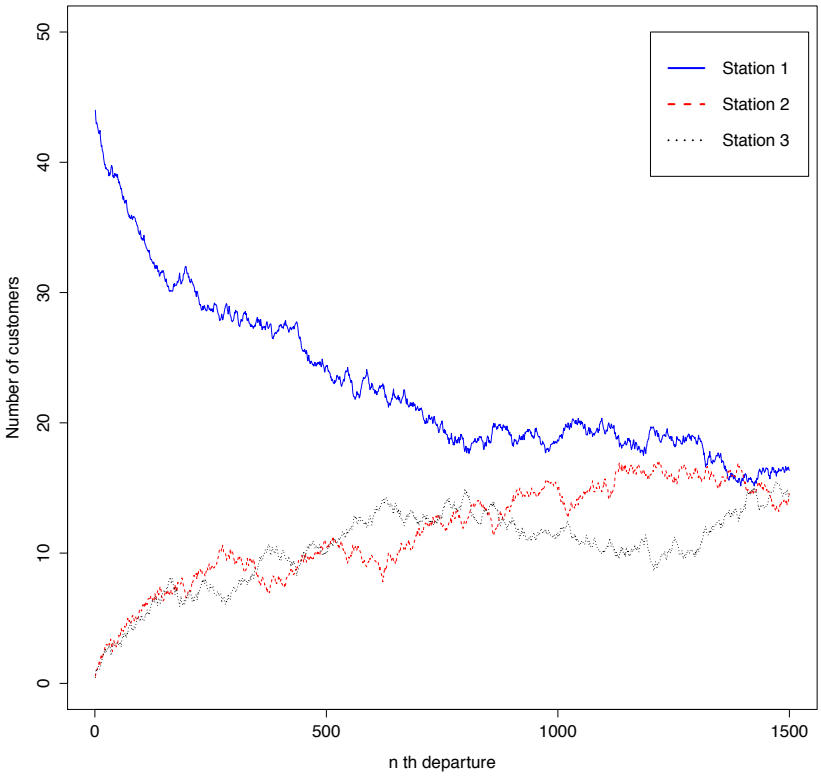

Figure 5.2: A trajectory of the number of customers in each station when the nth customer departure occurs

The truncation point t_{n_0} will be chosen so that the data after t_{n_0}, $(X_1(t_n), \ldots, X_k(t_n) : n \geq n_0)$, are collected for steady-state analysis. It is worthwhile to note that the simulation output is *multi-dimensional* because it records the numbers of customers in several stations. And this multidimensional case was not introduced nor studied previously in the literature.

Example 3. (s, S) **single item inventory model** An (s, S) inventory system is a widely used model in which the inventory position of a single item is reviewed periodically. At the end of each time period, if the inventory position is found to be below s units, additional units are ordered to bring the inventory position up to S. If the inventory position is found to be above s, then no additional units are ordered.

The following costs are usually considered: a fixed setup cost whenever an order is placed, an incremental cost per item ordered, a holding cost per unit per unit time, and a backlog cost per unit per unit time. Suppose that we wish to compute the long-run average cost per unit time at a fixed (s', S'), then we need to simulate the system, compute the total cost incurred in each time period, and obtain the arithmetic average of the costs over time. The initial inventory position is set arbitrarily, so the sequence of costs over time exhibits a transient behavior before it converges to its steady-state. Figure 5.3 shows a trajectory of the total costs over time in an (s, S) inventory system.

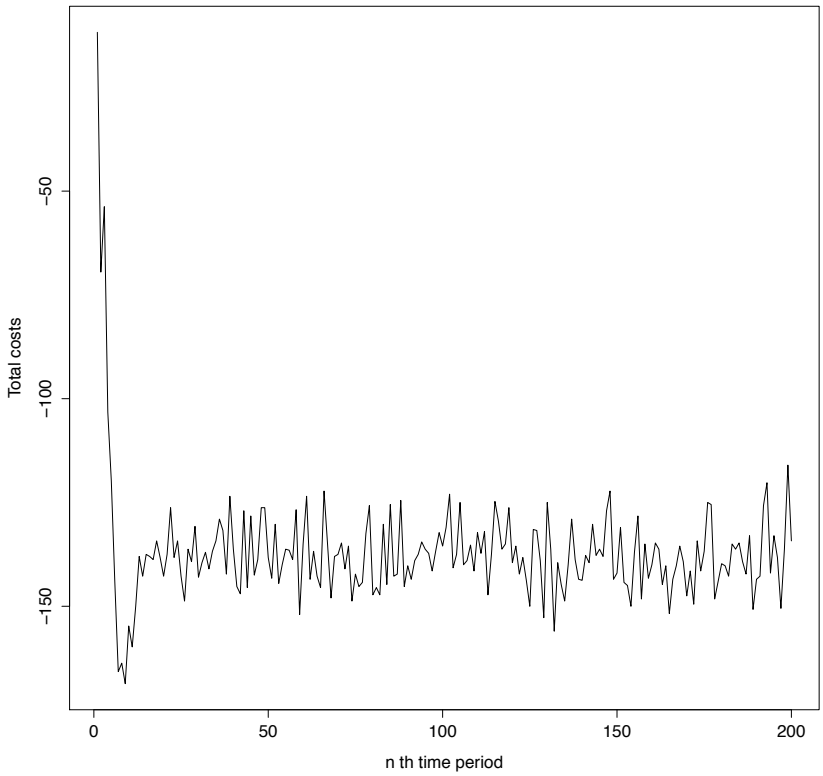

Figure 5.3: A trajectory of the total costs over time in an (s, S) inventory system

In the above examples, we consider the problem of computing the steady-state mean of a system performance in the situation where it is not clear how to initialize

the simulation and hence the simulation must start at some arbitrary position which is usually atypical of steady-state behavior. This particular initialization will induce an initial transient phase that introduces a severe bias in the point and interval estimates. And this bias has great effect on our simulation-based optimization algorithms since most algorithms rely on point or interval estimates to update the searching direction and decide the feasibility. To overcome this problem, we will allow the system to warm up before output data are collected and eliminate the observations until steady-state behavior becomes apparent. We then compute the arithmetic mean of the remaining observations as an estimate of the steady-state mean.

5.1.2 General Procedures

In our proposed method, the truncation point after which the observations are retained for steady-state analysis will be chosen according to the following technique:

1. Divide the simulation output into small batches.

2. Obtain the empirical distribution function within each batch.

3. Compare these empirical distribution functions for a change in distribution functions.

The main idea underlying the proposed methodology is that, in a stationary process, the distribution in one time period is same as the distribution in the next time period. So, we want to find a point in the simulation output where the distribution in one time period is not significantly different from the distribution of the next time period. Hence after batching the simulation output into several batches, we treat each batch as one period of time and treat the observations within each batch as if they follow a common distribution. Then the empirical distribution functions are obtained from each batch and we consider them representatives of the distribution of each batch. Assuming that the simulation has evolved long enough so that the last

batch is in steady-state, we compare the empirical distribution function from the first batch to that of the last batch and if the two are significantly different, we conclude that the first batch is in the transient phase. We then move to the second batch and compare its empirical distribution function to that of the last batch and see if the two are significantly different. The procedure is repeated until we find a batch whose empirical distribution function is not significantly different from that of the last batch. Observations after this batch are retained for steady-state analysis. The graphical representation of the proposed procedure is displayed in Figure 5.4.

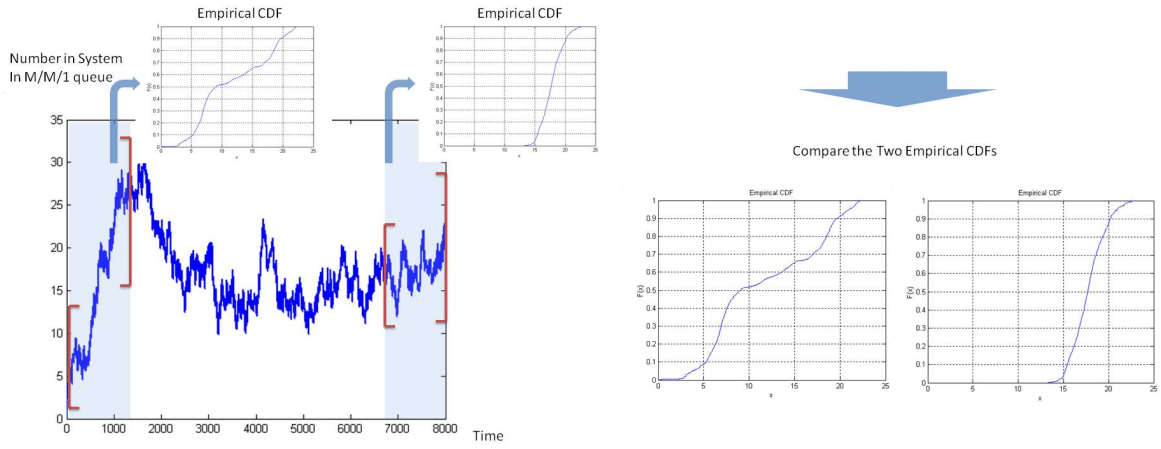

Figure 5.4: Graphical representation of the proposed procedure

The proposed method is rigorously described in Section 5.2 and its efficiency is demonstrated in Section 5.3.

5.2. Analysis Framework

We consider a d-dimensional stochastic process $(Y_1, \ldots, Y_n)$ which represents the output of a discrete event simulation. We wish to compute $\mu \triangleq \lim_{n\to\infty} EY_n$ through simulation. A natural estimator for μ is the sample mean $\sum_{i=1}^{n} Y_i/n$. Another point estimator is the truncated sample mean $\sum_{i=t_0+1}^{n} Y_i/(n-t_0)$, which is the mean of

the reserved series, $(Y_i : i = t_0 + 1, \ldots, n)$, after the first t_0 observations have been deleted.

5.2.1 Kolmogorov-Smirnov Test

In our proposed method, a Kolmogorov-Smirnov test is used to test if the simulation output observations from two different batches follow a common distribution. In statistics, the Kolmogorov-Smirnov test is a commonly used nonparametric test technique to determine if data from two samples differ significantly. It measures the distance between the empirical distribution functions of the two samples. The null hypothesis in the test is that there is no significant different between the distribution of the two samples.

The Kolmogorov-Smirnov test compares the empirical cumulative distribution function (cdf), $F^1(x)$, of the first sample with the empirical cdf, $F^2(x)$, of the second sample. Here we assume the two sample sizes are equal. By definition, for a particular data sample i with data $Y_1, Y_2, \ldots, Y_n$, the empirical cdf $F^i(x)$ can be computed by

$$F^i(x) = \frac{\text{number of } Y_1, Y_2, \ldots, Y_n \text{ which are } \leq x}{n}$$

The cdf of an empirical distribution is a step function which jumps at each observed data point. Figure 5.5 shows a typical behavior of an empirical cdf.

After obtaining the empirical cdf of the two samples, the Kolmogorov-Smirnov test statistic is computed as the largest absolute deviation between $F^1(x)$ and $F^2(x)$ over the range of the random variable. Denote the test statistic as $D(F^1, F^2)$,

$$D(F^1, F^2) = \sup |F^1(x) - F^2(x)|. \tag{5.1}$$

See Figure 5.6 for an illustration to compute D. The sample distribution of Kolmogorov-Smirnov test statistic D is known; the table of critical values can be found at Banks

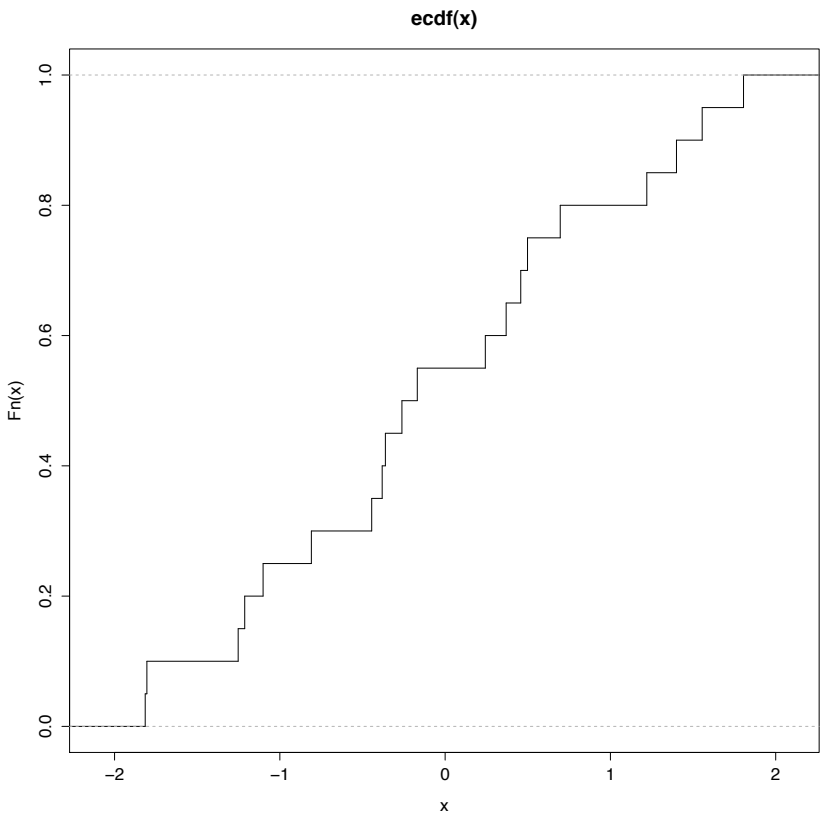

Figure 5.5: Empirical cumulative distribution function

et al. (2001). If the sample statistic D is greater than the critical value, D_α, with significance level α and size n, the null hypothesis that the two data samples follow the same distribution is rejected.

Note that we only consider the one-dimensional case in the above Kolmogorov-Simirnov test. If $d \geq 2$, the method mentioned above might not be appropriate to compute the test statistic based on the maximum distance of the empirical cumulative distribution function. In one-dimensional case, the test statistic is independent of the direction of counting the frequency since $P(x \leq X) = 1 - P(x > X)$. However, when it comes to the d-dimensional case, we need to consider 2^d independent ways of defining a cumulative distribution function. Taking $d = 2$ for example, given the 2-dimensional simulation output data (X_i, Y_i), $i = 1, 2, \ldots, n$, the empirical cumulative

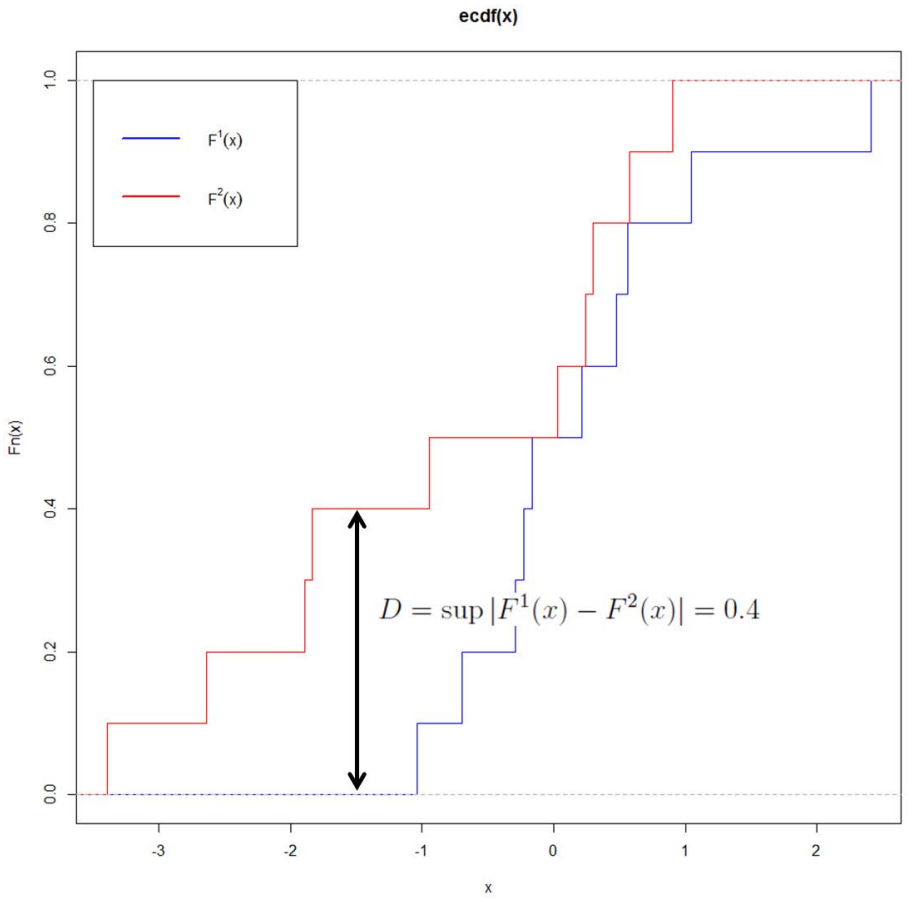

Figure 5.6: Computing Kolmogorov-Smirnov test statistic

distribution function should be defined by considering 4 different directions: $(x \leq X, y \leq Y)$, $(x \leq X, y > Y)$, $(x > X, y \leq Y)$, and $(x > X, y > Y)$; see Peacock (1983). As a result, for each data point (X_i, Y_i), $i = 1, 2, \ldots, n$, we should count the frequencies of points in all 4 quadrants of the plane: $(x \leq X_i, y \leq Y_i)$, $(x \leq X_i, y > Y_i)$, $(x > X_i, y \leq Y_i)$, and $(x > X_i, y > Y_i)$. Then for each direction, we calculate the maximum distance of the cumulative distribution function between the two samples and the maximum distance among the four directions is selected as the test statistic.

There are several advantages of using Kolmogorov-Smirnov test for simulation output analysis. First of all, Kolmogorov-Smirnov test doesn't rely any assumptions on the data distribution. Secondly, Kolmogorov-Smirnov test is a exact test. It

is sensitive to both location and shape of the empirical cdf of the data samples. But many other statistical tests used in the warm-up detection only compare the means, which might reduce the accuracy if the data are not normally distributed. Furthermore, the procedures to compute Kolmogorov-Smirnov test statistic is quite simple, and can be automated in most programming languages. Many statistical software packages such as Matlab, R, and SAS include functions or procedures to conduct Kolmogorov-Smirnov test. Therefore, Kolmogorov-Smirnov test is a very promising technique to apply for initial transient period detection.

5.2.2 A Kolmogorov-Smirnov Test based Algorithm

The algorithm to determine the truncation point, t_0, is summarized in the proposed methodology.

Proposed Methodology

---

1. Run the simulation for n time units. Repeat this m times independently. Let $Y_i(j) \in \mathcal{R}^d$ be the ith observation from the jth replication. So, $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

2. Compute the averages over the replications. Set

$$\overline{Y}_i = \sum_{j=1}^{m} \frac{Y_i(j)}{m}$$

for $i = 1, \ldots, m$. Note that a natural point estimator for μ is the untruncated sample mean:

$$Y_1^* = \frac{1}{n} \sum_{i=1}^{n} \overline{Y}_i. \tag{5.2}$$

3. The averages $(\overline{Y}_i : i = 1, \ldots, n)$ are batched in b batches of size $n_b = n/b$. For each batch, compute the empirical cumulative distribution function using the observations within each batch, i.e., the empirical distribution function of the kth batch is computed by:

$$F^k(x) = \frac{1}{n_b} \sum_{s=(i-1)n_b+1}^{in_b} I(\overline{Y}_s \leq x), \quad x \in \mathcal{R}^d, k = 1, \ldots, b,$$

where I denotes the indicator function.

4. Compare the last empirical distribution function, F^b, with the previous ones, $F^k, k = 1, \ldots, b-1$ for a change in distribution functions. For instance, we can compute the distance between each pair of distribution functions in the supremum norm,

$$D(F^b, F^k) = \sup_{x \in \mathcal{R}^d} |F^b(x) - F^k(x)| \tag{5.3}$$

for $k = 1, \ldots, b-1$ and check whether this quantity is significantly different from zero.

5. Set b_0 to be the first batch that $D(F^{b_0}, F^b)$ is below ϵ for some small $\epsilon > 0$. Set the truncation point t_0 to be the position of the first observation in batch b_0. Collect data after t_0 for steady-state analysis. The proposed estimator for μ is the truncated sample mean which is the arithmetic average of the observations after t_0:

$$Y^{**} = \frac{1}{n - t_0} \sum_{i=t_0+1}^{n} \overline{Y}_i. \tag{5.4}$$

5.3. Numerical Experiments

In this section, we will study the empirical performance of the proposed procedure using Examples 1, 2, and 3. We will compare the performance of the proposed estimator Y^{**} to that of the standard estimator Y_1^*, as well as another two estimators Y_2^*, Y_3^* discussed in Pawlikowski (1990).

Y_2^* is obtained based on the rule R4 of Pawlikowski (1990). The basic idea is that the initial transient period is over after k consecutive values of running mean $\overline{\overline{Y}}_i$ approach a constant level with a given accuracy δ. The running mean $\overline{\overline{Y}}_i$ at a particular point h is calculated as

$$\overline{\overline{Y}}_h = \sum_{i=1}^{h} \overline{Y}_i$$

To be specifically, if the running mean $\overline{\overline{Y}}_i$ after the observation t_0 differ less than $100\delta\%$ from $\overline{\overline{Y}}_{t_0+k}$, i.e., for all i, $t_0 < i \leq t_0 + k$,

$$\frac{|\overline{\overline{Y}}_{t_0+k} - \overline{\overline{Y}}_i|}{|\overline{\overline{Y}}_{t_0+k}|} < \delta,$$

then t_0 is the truncation point.

Y_3^* is calculated based on the rule R5 (crossing of the mean rule) of Pawlikowski (1990). According to this rule, the initial transient period is over after t_0 observations if the time series $Y_1, \ldots, Y_{t_0}$ crosses the running mean $\overline{\overline{Y}}_{t_0}$ l times.

Example 1 Revisited. We consider an M/M/1 queue with the service rate μ and the arrival rate λ under the first-come-first-served (FCFS) discipline. We wish to estimate the long-run average waiting time of customers in the queue. The system is initialized idle and empty. Note that the expected waiting time of the nth customer

$E(w_n)$ in this setting can be computed by

$$E(w_n) = \frac{1}{\mu}\sum_{i=2}^{n}(i-1)P_0(n,i),$$

where $P_0(n,i)$ is the probability that there are i customers present in the system when the nth customer arrives given 0 customer present at time 0 (Kelton and Law, 1985). And $P_0(n,i)$ can be calculated by the Algorithm 1 suggested in Kelton and Law (1985). We conducted two sets of experiments to compare the performance of the proposed method with the other two methods in detecting the initial transient period.

In the first experiment, we set $\lambda = 0.8$, $\mu = 1$, and the traffic density $\rho = 0.8$. The theoretical value of long-run average waiting time of customer in the queue is $w = \dfrac{\rho}{\mu(1-\rho)} = 4$. In Figure 5.7 we plot the convergence of $E(w_n)$ over time.

We define the theoretical truncation point as the smallest point beyond which $E(w_n)$ falls within 5% of w. Then the theoretical truncation point according to Figure 5.7 is 99. The parameters m, n, and b in the proposed algorithm are set to be $50, 2000$, and 10, respectively. The parameters k, δ in $R4$ are set to be 30 and 0.01, while l in $R5$ is set to be 25. Table 5.1 shows the bias, the variance and the mean squared error of Y_1^*, Y_2^*, Y_3^* and Y^{**} based on 20 independent samples. The last column of the table also reports the average truncation point t_0.

Table 5.1: Performance of Y_1^*, Y_2^*, Y_3^* and Y^{**} in Examples 1 with $\rho = 0.8$

	$\lvert\text{estimator} - w\rvert$	Variance	MSE	Avg. t_0
Y_1^*	0.1106	0.0124	0.0163	-
Y_2^*	0.1030	0.0131	0.0131	152
Y_3^*	0.1055	0.0156	0.0150	258
Y^{**}	0.0965	0.0122	0.0120	96

From Table 5.1 we can see that Y^{**} has the least bias, variance and MSE. And the average truncation point suggested by the proposed method is very closed to 99.

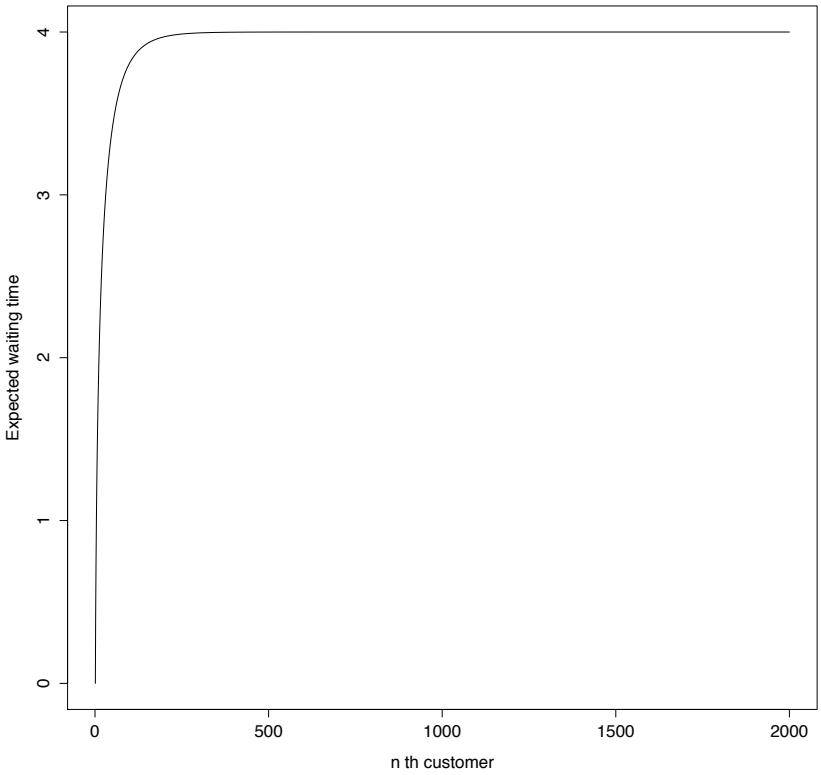

Figure 5.7: $E(w_n)$ as a function of n for the M/M/1 queue with $\rho = 0.8$

In the second experiment, we set $\lambda = 0.96$, $\mu = 1$, and the traffic density $\rho = 0.96$. The theoretical value of long-run average waiting time of customer in the queue is $w = \dfrac{\rho}{\mu(1-\rho)} = 24$. In Figure 5.8 we plot the convergence of $E(w_n)$ over time. The theoretical truncation point according to Figure 5.8 is 2634. The parameters m, n, and b in the proposed algorithm are set to be $50, 4000$, and 10, respectively. The parameters k, δ in $R4$ are set to be 10 and 0.01, while l in $R5$ is set to be 10. Table 5.2 shows the bias, the variance and the mean squared error of Y_1^*, Y_2^*, Y_3^* and Y^{**} based on 20 independent samples. The last column of the table also reports the average truncation point t_0.

Although Table 5.2 shows that the variance and MSE of Y^{**} are larger than the

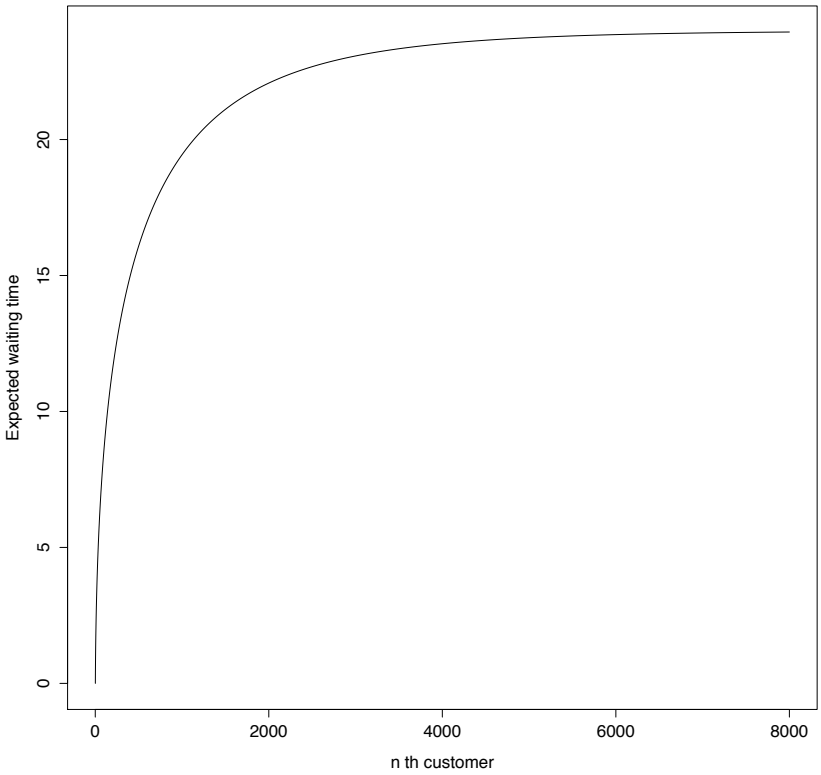

Figure 5.8: $E(w_n)$ as a function of n for the M/M/1 queue with $\rho = 0.96$

Table 5.2: Performance of Y_1^*, Y_2^*, Y_3^* and Y^{**} in Examples 1 with $\rho = 0.96$

	$\lvert\text{estimator} - w\rvert$	Variance	MSE	Avg. t_0
Y_1^*	3.3268	1.7570	12.7368	-
Y_2^*	2.4504	1.9892	7.8944	311
Y_3^*	2.6530	1.9815	8.9206	199
Y^{**}	2.0863	10.0212	9.5204	2217

other two methods due to the stochastic property of the observed data, the proposed method provides better estimate of the truncation point.

Example 2 Revisited. We consider a closed Jackson network with 3 stations. Suppose that a customer departing a station is routed to one of the two remaining stations with a probability of 0.5 for each. Assume that the total number of customers in the network is 45. Let $X = ((X_n^1, X_n^2, X_n^3) : n \geq 0)$ represent the numbers of

customers in stations 1, 2, and 3 respectively at the n'th epoch of the departure process of customers. Set $(X_0^1, X_0^2, X_0^3) = (45, 0, 0)$. We wish to estimate the long-run average number of customers in each station. m, n, b, k, δ, and l are set to be $50, 1500, 10, 50, 0.01$ and 10, respectively. It is necessary to note that the original $R4$ and $R5$ are only designed for one-dimension simulation output analysis. For this *three-dimensional* queueing network, we should make some modifications for the original rules. One simple and conservative way to handle this problem is to identify the truncation points for each dimension first, then the maximum value of these truncation points is chosen for the overall truncation point. Table 5.3 shows the bias, variance and the mean squared error of Y_1^*, Y_2^*, Y_3^* and Y^{**} based on 20 independent samples. Note $X^* = (15, 15, 15)$.

Table 5.3: Performance of Y_1^*, Y_2^*, Y_3^* and Y^{**} in Examples 2

	$\lvert\text{estimator} - X^*\rvert$	Variance	MSE
Y_1^*	$(7.75, 3.77, 3.99)$	$(1.73, 0.90, 1.11)$	$(31.74, 15.04, 16.94)$
Y_2^*	$(2.85, 1.85, 1.87)$	$(10.45, 7.11, 4.74)$	$(11.71, 6.88, 5.45)$
Y_3^*	$(4.79, 2.26, 2.65)$	$(2.41, 1.18, 2.07)$	$(25.23, 6.11, 8.50)$
Y^{**}	$(2.35, 1.91, 1.34)$	$(6.74, 7.40, 2.80)$	$(6.63, 7.04, 2.97)$

Table 5.3 shows that Y^{**} computed by the proposed method has the smallest bias and MSE.

Example 3 Revisited. We consider (s, S) inventory system with $s = S = 22$ where the demands $D_1, D_2, \ldots$ are independent identically distributed random variables with the common probability mass function $P(D_1 = 10) = 0.1, P(D_1 = 15) = 0.1, P(D_1 = 20) = 0.4, P(D_1 = 25) = 0.3, P(D_1 = 30) = 0.1$. We assume that the lead time of an order is zero. The inventory position is defined as the stock on hand plus that on order minus the backlogged. The fixed setup cost is \$0 and the incremental cost is $c = \$3$ per item ordered. The holding cost is $h = \$1$ per unit per unit time and the backlog cost is $p = \$4$ per unit per unit time. Suppose that each unit

is sold for $a = \$10$. We wish to compute the long-run average cost per period. Note that the actual long-run average cost per period is given by

$$(c-a)ED_1 + hE\max(S - D_1, 0) + pE\max(D_1 - S, 0),$$

which is $-\$137.5$. See page 285 of Nahmias (2005) for more detail. The initial inventory position is set to be 100. m, n, b, k, δ, and l are set to be $20, 100, 5, 30, 0.01$ and 25, respectively. Table 5.4 shows the bias and the mean squared error of Y_1^*, Y_2^*, Y_3^* and Y^{**} based on 20 independent samples.

Table 5.4: Performance of Y_1^*, Y_2^*, Y_3^* and Y^{**} in Examples 3

	$\lvert\text{estimator} - \mu\rvert$	Variance	MSE
Y_1^*	2.4076	0.6002	6.3668
Y_2^*	1.0380	1.6351	1.5707
Y_3^*	1.5025	3.6298	3.6183
Y^{**}	0.7591	0.9329	0.9087

Table 5.4 shows that Y^{**} beats the other three with the least bias, variance and MSE.

Table 5.1, 5.2, 5.3 and 5.4 present the effectiveness of the proposed estimator, Y^{**}, comparing with the standard estimator, Y_1^* and the other estimators Y_2^*, Y_3^* in terms of $\lvert\text{estimator} - \mu\rvert$, the variance, and the mean squared error. One of the advantages of the proposed estimator over other alternatives is that the performance of the proposed estimator is not sensitive to the parameter b, which is the number of batches in each simulation output. However, to implement Y_2^* and Y_3^*, more attention should be paid to the parameters k, δ, and l according for each specific situation. Furthermore, the system-dependent selection of these parameters seems to be too arduous for users sometimes. From this point of view, the proposed method is further recommended.

It is worthwhile to note that the simulation output is *multi-dimensional* in Example 2 and the proposed methodology performs efficiently in that case as well.

Chapter 6

Conclusion

In this dissertation, we first propose a novel method for simulation-based optimization over discrete sets in the presence of stochastic constraints. The key idea of the proposed method is to convert constrained optimization into an unconstrained problem of finding a saddle point of the Lagrangian. This approach is motivated by the difficulty in identifying a feasible solution in the presence of stochastic constraints. The proposed approach is a.s. convergent to the optimal solution under appropriate conditions and it displays good performance in comparison with other competing methods, as illustrated through numerical experiments.

Then we study the problem of fitting a convex function based on noisy simulation output data. Traditionally, the convex function is computed by minimizing the sum of least squares, which is proven to be time-consuming when working with large dataset. It might also run out of the computer memory since the formulation has too many constraints. In Chapter 4, we propose a computationally efficient way to fit the convex function by minimizing the sum of least absolute deviations instead of the sum of squares. The proposed least absolute deviations formulation can be easily converted to a linear program. Furthermore, the LP formulation has a dual problem that exhibits a block-angular structure in its constraints, which enables one to apply

Dantzig-Wolfe decomposition techniques to solve it efficiently. We present several numerical examples to illustrate that the proposed estimator can be computed faster and for larger datasets than the traditional least squares estimator. We also establish the consistency of the proposed estimator both numerically and theoretically.

The last part of the dissertation discusses the initial transient period detection techniques in the framework of simulation-based optimization. We present a simple yet efficient statistical test based technique to detect the warm-up period automatically in discrete-event simulations. The statistical test is conducted by comparing the empirical distribution function of the data samples. We divide the simulation output time-series into small batches, and then compare the empirical distribution function from the first batch to that of the last batch. If the test shows statistical significance, we conclude that the first batch is in the transient phase and the data in this batch need to be deleted. Then, we move to the second batch and repeat the procedure until we find a batch whose empirical distribution function is not significantly different from that of the last batch. The proposed methodology can be viewed as a generalization of traditional truncation techniques for that it can deal with *multi-dimensional* state variable. The numerical experiments show that the suggested method achieves the best performance in terms of $|\text{estimator} - \mu|$, the variance, and the mean squared error. Moreover, the performance of the proposed method doesn't rely too much on the chosen parameters, while other truncation techniques heavily depend on pre-specified parameters.

While we have seen some promising results of the proposed methods in simulation-base optimization, convex regression, and initial transient period detection, the following research directions could be explored to improve our proposed methods and procedures.

For the simulation-based optimization methods that handle noisy constraints and discrete decision variables in Chapter 3, we might consider the following alternatives:

- Choose step size sequence. Although the step size sequence c_n only needs to satisfy $\sum_{n=1}^{\infty} c_n = \infty$ and $\sum_{n=1}^{\infty} c_n^2 \leq \infty$ to guarantee the SA procedure to converge, the choice of c_n does affect the performance of the proposed method. If c_n is too small, θ_n proceeds very slowly and cannot achieve the expected performance with a given simulation budget. On the other hand, if c_n is too large, the algorithm becomes unstable and cannot converge to the optimal solution. In the illustrative example, we use $c_n = c/n$ as the step size sequence. However, there is uncertainty regarding whether this type of sequences will work well in other optimization problems or not. Hence it is valuable to investigate if there are any adaptive step sequences which can be used in the proposed method.

- Use common random numbers. In the proposed method, we assume that each simulation run uses independent stream of random numbers. To provide better estimates of the subgradient in each iteration, we might consider using common random numbers (CRN) to run simulations at the $d + 1$ different designs. By using the same streams of random numbers on different simulation scenarios, we might reduce the variances when estimating the mean performance differences between these designs.

- Integrate the proposed method with other optimization techniques. Some heuristics in the simulation-based optimization literature might be used to get the initial solutions. Then we may apply the proposed method and start with several different initial candidates. And the solution with the best performance can be used as the final optimal solution. Meanwhile, we can also apply our proposed method within the frame work of optimal computing budget allocation to spend simulation effort smartly.

With regard to the least absolute deviations formulation to compute the convex regression estimators, our research could be extended in the following directions:

- Develop more efficient algorithm to solve (4.4). Although we demonstrate the efficiency of applying Dantzig-Wolfe decomposition techniques to solve (4.4), there are many alternative ways to improve the algorithm. For example, in each iteration of the decomposition algorithm, we need to solve a series of subproblems. The current method employs simplex algorithm to find the optimal solutions of these small linear programs. However, the convergency of the Dantzig-Wolfe decomposition algorithm does not require us to solve each subproblem to optimality. As long as we can find a feasible solution of the subproblem with positive reduced cost, it can be entered into the master problem. From this point of view, we can take advantage of the special structure of the subproblems and develop heuristics to find candidate solutions quickly.

- Apply the convex regression technique in simulation-based optimization. The proposed convex regression estimator has provable consistency and enjoys computational advantages. Therefore, we can apply it to construct convex response surface based on noisy simulation output data. Since the response surface is convex, many efficient optimization algorithms can be used to find the optimal solution of the approximated convex function.

In order to improve the performance of the Kolmogorov-Simirnov test on detecting the warm-up period for steady-state simulation, we could further study the following directions:

- Sample data from steady-state distribution. In the proposed method, we use the data from the final batch as the steady-state data. This choice is based on the the assumption that the simulation has been running long enough to reach the steady state. However, it is not always the case for all steady-state simulations. Even though the simulation has been running long enough, the data from the final batch might be highly biased due to the stochastic property

of the system. As a result, we need to design a way to ensure that the data of the final batch are indeed from the steady-state distribution. Alternatively, we might develop some procedures to sample data from the simulation output and form a new batch. We can conduct some statistical tests to make sure the new batch has the same distribution as the steady-state. Then, this batch will be used by our proposed method to find the truncation point.

- Develop efficient algorithms to compare the empirical distribution function in multi-dimensional case. Although we explain the basic idea of extending the one-dimensional Kolmogorov-Simirnov test to multi-dimensional simulation output data, the method of computing the empirical distribution function is not very efficient. Given two d-dimensional data samples with size n_1 and n_2, the method requires computing the cumulative frequencies for each of the $(n_1+n_2)2^d$ subspace. That could be computationally expensive when d or sample size is very large. So efficient algorithm is needed to compute the empirical distribution function in multi-dimensional case and for large dataset.

Bibliography

Aguilera, N., Forzani, L., and Morin, P. (2011). On uniform consistent estimators for convex regression. *Journal of Nonparametric Statistics*, 23(4):897–908.

Ahmed, M. A., Alkhamis, T. M., and Hasan, M. (1997). Optimization discrete stochastic systems using simulated annealing and simulation. *Computer and Industrial Engineering*, 32(4):823–836.

Alkhamis, T. M. and Ahmed, M. A. (2005). Simulation-based optimization for repairable systems using particle swarm algorithm. In *Proceedings of the 2005 Winter Simulation Conference*, pages 857–861.

Allon, G., Beenstock, M., Hackman, S., Passy, U., and Shapiro, A. (2007). Nonparametric estimation of concave production technologies by entropic methods. *Journal of Applied Econometrics*, 22(4):795–816.

Alrefaei, M. and Andradï¿½ttir, S. (1999). A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science*, 45:748–764.

Alrefaei, M. and Andradï¿½ttir, S. (2001). A modification of the stochastic ruler method for discrete stochastic optimization. *European Journal of Operational Research*, 133:160–182.

Andradóttir, S. (1995). A method for discrete stochastic optimization. *Management Science*, 41(12):1946–1961.

Andradóttir, S., Goldsman, D., and Kim, S.-H. (2005). Finding the best in the presence of a stochastic constraint. In Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 732–738, Piscataway, New Jersey. Institute of Electrical and Electronics Engineers, Inc.

Andradóttir, S. and Kim, S. H. (2010). Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics*, 57:403–421.

Andradï¿½ttir, S. (1996). A global search method for discrete stochastic optimization. *SIAM Journal on Optimization*, 6:513–530.

Andradï¿½ttir, S. (1998). A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference*, pages 151–158.

Andradï¿½ttir, S. (1999). Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, 9:349–380.

Andradï¿½ttir, S. (2006). An overview of simulation optimization via random search. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 20, pages 617 – 631. Elsevier.

Ankenman, B., Nelson, B. L., and Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382.

April, J., Glover, F., Kelly, J., and Laguna, M. (2003). Practical introduction to simulation optimization. In Chick, S., Sï¿½nchez, P. J., Ferrin, D., and Morrice., D. J., editors, *Proceedings of the 2003 Winter Simulation Conference*, pages 71–78, Piscataway, New Jersey. Institute of Electrical and Electronics Engineers, Inc.

Banks, J., Carson, J. S., Nelson, B. L., and Nicol, D. M. (2001). *Discrete-event System Simulation*. Prentice-Hall, New Jersey, 4th edition.

Barton, R. R. (1992). Metamodels for simulation input-output relations. In *Proceedings of the 1992 Winter Simulation Conference*, pages 289–299, New York, NY, USA. ACM.

Barton, R. R. (2009). Simulation optimization using metamodels. In Rossetti, M. D., Hill, R. R., Johansson, B., Dunkin, A., and Ingalls, R. G., editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 230–238, Piscataway, New Jersey. Institute of Electrical and Electronics Engineers, Inc.

Barton, R. R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 18, pages 535 – 574. Elsevier.

Bassett, G. and Koenker, R. (1978). Asymptotic theory of least absolute error regression. *Journal of the American Statistical Association*, 73(363):618–622.

Batur, D. and Kim, S. H. (2001). Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Transactions on Modeling and Computer Simulation*, 47:800–816.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2006). *Nonlinear programming: Theory and Algorithms*. John Wiley & Sons, Inc., New Jersey.

Benveniste, A., Métivier, M., and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer–Verlag, New York.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA.

Birke, M. and Dette, H. (2007). Estimating a convex function in nonparametric regression. *Scandinavian Journal of Statistics*, 34(2):384–404.

Boesel, J., Nelson, B. L., and Kim, S.-H. (2003). Using ranking and selection to clean up after simulation optimization. *Operations Research*, 51:814–825.

Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V., and Trosset, M. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17:1–13.

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, 13(1):1–45.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.

Bradley, S. P., Hax, A. C., and Magnanti, T. L. (1977). *Applied mathematical programming*. Addison-Wesley Pub. Co.

Bronshtein, E. M. (1976). ϵ–entropy of convex sets and functions. *Siberian Math. J.*, 17:393–398.

Carson, Y. and Maria, A. (1997). Simulation optimization: methods and applications. In *Proceedings of the 1997 Winter Simulation Conference*, pages 118–126, Washington, DC, USA.

Chang, I.-S., Chien, L.-C., Hsiung, C. A., C.-C.Wen, and Wu, Y.-J. (2007). Shape restricted regression with random bernstein polynomials. *Lecture Notes-Monograph Series*, 54:187–202.

Conway, R. W. (1963). Some tactical problems in digital simulation. *Management Science*, 10(1):47–61.

Dantzig, G. B. and Wolfe, P. (1961). The decomposition algorithm for linear programs. *Econometrica*, 29(4):767–778.

del Mar Hershenson, M., Boyd, S. P., and Lee, T. H. (2001). Optimal design of a CMOS opamp via geometric programming. *IEEE Trans. on Computer–Aided Design of Integrated Circuits and Systems*, 20(1):1–21.

Dent, W. (1973). A note on least squares fitting of functions constrained to be either nonnegative, nondecreasing or convex. *Management Science*, 20(1):130–132.

du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194:229–237.

Dupač, V. and Herkenrath, U. (1983). Stochastic approximation on a discrete set and the multi–armed bandit problem. *Communications in Statistics–Sequential Analysis*, 1:1–25.

Fishman, G. S. (1973). *Concepts And Methods In Discrete Event Digital Simulation*. Wiley, New York.

Fraser, D. A. S. and Massam, H. (1989). A mixed primal-dual bases algorithm for regression under inequality constraints: Application to concave regression. *Scandinavian Journal of Statistics*, 16:65–74.

Fu, M. and Hu, J. Q. (1997). *Conditional Monte Carlo: gradient estimation and optimization applications*. Kluwer Academic, Boston.

Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215.

Fu, M. C. (2006). Gradient estimation. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 19, pages 575 – 616. Elsevier.

Fu, M. C. (2008). What you should know about simulation and derivatives. *Naval Research Logistics*, 55(8):723–736.

Fu, M. C. and Hill, S. D. (1997). Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions*, 29(3):233–243.

Gafarian, A. V., Jnr, C. J. A., and Morisaku, T. (1978). Evaluation of commonly used rules for detecting steady state in computer simulation. *Naval Research Logistics Quarterly*, 25:511–529.

Gelfand, S. B. and Mitter, S. K. (1989). Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications*, 62(1):49–62.

Gerencsér, L. S., Hill, D., and Vago, Z. (1999). Optimization over discrete sets via spsa. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1791–1795.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166.

Glover, F. (1989). Tabu search–part i. *ORSA Journal on Computing*, 1:190–206.

Glover, F. and Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Gokbayrak, K. and Cassandras, C. G. (2002). Generalized surrogate problem methodology for online stochastic discrete optimization. *Journal of Optimization Theory and Applications*, 114(1):97–132.

Goldsman, D. and Nelson, B. L. (1994). Ranking, selection and multiple comparisons in computer simultaions. In *Proceedings of the 1994 Winter Simulation Conference*, pages 192–199.

Goldsman, D. and Nelson, B. L. (1998). *Comparing Systems via Simulation*, pages 273–306. John Wiley & Sons, Inc.

Goldsman, D., Schruben, L. W., and Swain, J. J. (1994). Tests for transient means in simulated time series. *Naval Research Logistics*, 41:171–187.

Gong, W.-B., Ho, Y.-C., and Zhai, W. (1999). Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*, 10:384–404.

Gordon, G. (1969). *System Simulation*. Prentice-Hall, New Jersey.

Grover, W. D. (2004). *Mesh-Based Survivable Networks: Options and Strategies for Optical, Mpls, Sonet, and Atm Networking*. Prentice Hall.

Hannah, L. A. and Dunson, D. B. (2011). Bayesian nonparametric multivariate convex regression.

Hanson, D. L. and Pledger, G. (1976). Consistency in concave regression. *Ann. Statist.*, 4(6):1038–1050.

Henderson, S. G. and Nelson, B. L. (2006). *Hanbooks in Operations Research and Management Science*, volume 13 of *Simulation*. Elsevier, Amsterdam, The Netherlands.

Hildreth, C. (1954). Point estimates of ordinates of concave functions. *Journal of the American Statistical Association*, 49(267):598–619.

Hill, D., Gerencsér, L., and Vágó, Z. (2003). Stochastic approximation on discrete sets using simultaneous perturbation difference approximations. In *Conference on Information Science and Systems*.

Ho, J. K. and Loute, E. (1981). An advanced implementation of the dantzig-wolfe decomposition algorithm for linear programming. *Mathematical Programming*, 20(1):303–326.

Ho, Y. C., S. R. and Vakili, P. (1992). Ordinal optimization of discrete event dynamic systems. *Journal of Discrete Event Dynamic Systems*, 2(2):61–88.

Ho, Y. C. and Cao, X. R. (1983). Perturbation analysis and optimization of queueing networks. *Journal of Optimization Theory and Applications*, 40:559–582.

Hoad, K., Robinson, S., and Davies, R. (2008). Automating warm-up length estimation. In *Proceedings of the 2008 Winter Simulation Conference*, pages 532–540.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Hong, L. J. and Nelson, B. L. (2006). Discrete optimization via simulation using compass. *Operations Research*, 54(1):115–129.

Hong, L. J. and Nelson, B. L. (2009). A brief introduction to optimization via simulation. In Jain, S., Creasey, R. R., Himmelspach, J., White, K. P., and M. Fu, e., editors, *Proceedings of the 2009 Winter Simulation Conference*, pages 75–85, Piscataway, New Jersey. Institute of Electrical and Electronics Engineers, Inc.

Hong, L. J., Nelson, B. L., and Xu, J. (2010). Speeding up compass for high-dimensional discrete optimization via simulation. *Operations Research Letters*, 38:550–555.

Ishibuchi, H. and Murata, T. (1996). Multi-objective genetic local search algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 119–124, Piscataway, NJ,. IEEE Press.

Kelton, W. D. and Law, A. M. (1983). A new approach for dealing with the startup problem in discrete event simulation. *Naval Research Logistics Quarterly*, 30:641–658.

Kelton, W. D. and Law, A. M. (1985). The transient behavior of the m/m/s queue, with implications for steady-state simulation. *Operations Research*, 33(2):378–396.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, volume 4, pages 1942 –1948.

Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers, San Francisco, CA.

Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):pp. 462–466.

Kim, S.-H. and Nelson., B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11:251–273.

Kim, S.-H. and Nelson, B. L. (2006). Selecting the best system. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 17, pages 501 – 534. Elsevier.

Kleijnen, J. (2008a). Response surface methodology for constrained simulation optimization: An overview. *Simulation Modelling Practice and Theory*, 16:50–64.

Kleijnen, J. P. C. (1975). A comment on blanning's "metamodel for sensitivity analysis: the regression metamodel in simulation". *Interfaces*, 5(3):21 – 23.

Kleijnen, J. P. C. (2008b). *Design and Analysis of Simulation Experiments*. Springer Publishing Company, Incorporated, 1st edition.

Kleywegt, A. J., Shapiro, A., and Homem-de mello, T. (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

Kuosmanen, T. (2008). Representation theorem for convex nonparametric least squares. *Econometrics Journal*, 11(2):308–325.

Kurdila, A. J. and Zabarankin, M. (2005). *Convex Functional Analysis (Systems & Control: Foundations & Applications)*. Birkhäuser, Switzerland.

Kushner, H. and Yin, G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer.

Kushner, H. J. and Clark, D. C. (1978). *Stochastic approximation methods for constrained and unconstrained systems*. Springer–Verlag, New York.

Kushner, H. J. and Sanvicente, E. (1975). Stochastic approximation of constrained systems with system and constraint noise. *Automatica*, 11:375–380.

Law, A. M. (1983). Statistical analysis of simulation output data. *Operations Research*, 31:983–1029.

Law, A. M. and Kelton, W. D. (2000). *Simulation Modeling and Analysis*. McGraw-Hill, New York.

Lï¿½bbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Li, J., Sava, A., and Xie, X. (2009). Simulation-based discrete optimization of stochastic discrete event systems subject to non closed-form constraints. *IEEE Transactions on Automatic Control*, 54(12):2900–2904.

Liepins, G. E. and Hilliard, M. R. (1989). Genetic algorithms: foundations and applications. *Annals of Operations Research*, 21:31–58.

Lim, E. (2010). Response surface computation via simulation in the presence of convexity. In *Proceedings of 2010 Winter Simulation Conference*, pages 1246–1254.

Lim, E. and Glynn, P. W. (2012). Consistency of multidimensional convex regression. *Operations Research*, 60(1):196–208.

Luenberger, D. G. (1969). *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York.

Marsden, J. E. and Hoffman, M. J. (1993). *Elementary Classical Analysis*. W. H. Freeman and Company, New York.

Marsten, R. E., Hogan, W. W., and Blankenship, J. W. (1975). The boxstep method for large-scale optimization. *Operations Research*, 23(3):389–406.

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *Ann. Appl. Stat.*, 2(3):1013–1033.

Meyer, R. and Pratt, J. (1968). The consistent assessment and fairing of preference functions. *Systems Science and Cybernetics, IEEE Transactions on*, 4(3):270–278.

Mitchell, T. and Morris, M. (1992). The spatial correlation function approach to response surface estimation. In Swain, J., Goldsman, D., Crain, R., and Wilson, J., editors, *Proceedings of the 1992 Winter Simulation Conference*, pages 565–571, Piscataway, NJ. IEEE Press.

Murray, W. (1967). Ill–conditioning in barrier and penalty functions arising in constrained nonlinear programming. In *Proceedings of the Sixth International Symposium on Mathematical Programming*.

Nelson, B. L. (1992). Statistical analysis of simulation results. In *Handbook Of Industrial Engineering*. John Wiley, New York, 2nd edition.

Nelson, B. L. (2010). Optimization via simulation over discrete decision variables. In Hasenbein, J. J., editor, *TutORials in Operations Research Risk and Optimization in an Uncertain World*, pages 193–207. INFORMS.

Nelson, B. L. and Matejcik, F. J. (1995). Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Science*, 41:1935–1945.

Nelson, B. L., Swann, J., Goldsman, D., and Song, W. (2001). Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research*, 49:950–963.

Ólafsson, S. (2006). Metaheuristics. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 21, pages 633 – 654. Elsevier.

Ólafsson, S. and Kim, J. (2002). Simulation optimization. In *Proceedings of the 2002 Winter Simulation Conference*, pages 79 –84.

Paul, R. and Chanev, T. (1998). Simulation optimisation using a genetic algorithm. *Simulation Practice and Theory*, 6(6):601–611.

Paulson, E. (1964). A sequential procedure for selecting the population with the largest mean from k normal populations. *Annals of Mathematical Statistics*, 35:174–180.

Pawlikowski, K. (1990). Steady-state simulation of queueing processes: a survey of problems and solutions. *Computing Surveys*, 122(2):123–170.

Peacock, J. A. (1983). Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices Royal Astronomy Society*, 202:615–627.

Pichitlamken, J., Nelson, B. L., and Hong, L. J. (2006). A sequential procedure for neighborhood selection-of-the-best in optimization via simulation. *European Journal of Operational Research*, 173:283–298.

Prudius, A. A. and Andradóttir, S. (2009). Balanced explorative and exploitative search with estimation for simulation optimization. *INFORMS Journal on cmoputing*, 21:193–208.

Pujowidianto, N. A., Lee, L. H., Chen, C. H., and Yap, C. M. (2009). Optimal computing budget allocation for constrained optimization. In *Proceedings of the 2009 Winter Simulation Conference*, pages 584–589.

Reiman, Martin I.Weiss, A. (1989). Sensitivity analysis for simulations via likelihood ratios. *Operations Research*, 37(5):830.

Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics*, 7:799–811.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):pp. 400–407.

Roberts, A. W. and Varberg, D. E. (1974). Another proof that convex functions are locally Lipschitz. *Amer. Math. Monthly*, 81:1014–1016.

Robinson, S. (2002). A statistical process control approach for estimating the warm-up period. In *Proceedings of the 2002 Winter Simulation Conference*, pages 439–446.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton, New Jersey.

Roth, E. (1994). The relaxation time heuristic for the initial transient problem in m/m/k queueing systems. *European Journal of Operational Research*, 72:376–386.

Schruben, L., Singh, H., and Tierney, L. (1983). Optimal tests for initialization bias in simulation output. *Operations Research*, 31(6):1167–1178.

Schruben, L. W. (1982). Detecting initialization bias in simulation output. *Operations Research*, 30(3):569–590.

Seijo, E. and Sen, B. (2011). Nonparametric least squares estimation of a multivariate convex regression function. *Ann. Statist.*, 39:1633–1657.

Shanthikumar, J. G. and Yao, D. D. (1991). Strong stochastic convexity: closure properties and applications. *J. Appl. Prob.*, 28:131–145.

Shi, L. and Ólafsson, S. (2000). Nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability*, 2(3):271–291.

Shively, T. S., Walker, S. G., and Damien, P. (2011). Nonparametric function estimation subject to monotonicity, convexity and other shape constraints. *Journal of Econometrics*, 161(2):166–181.

Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization. In *7th Symposium on Multidisciplinary Analysis and Optimization*, AIAA-98-4755, pages 381–391, St. Louis, MO.

Skiba, A. K. (1978). Optimal growth with a convex-concave production function. *Econometrica*, 46(3):527–539.

Song, J. S., Wang, M., and Zhang, H. (2008). On the convexity of discrete (r, q) and (s, t) inventory systems. Available via https://faculty.fuqua.duke.edu/jssong/bio/Publications/Discrete-Convexity-08-Mar-31.pdf.

Spall, J. (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):817 –823.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341.

Spall, J. C. (1999). Stochastic optimization and the simultaneous perturbation method. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future - Volume 1*, WSC '99, pages 101–109, New York, NY, USA. ACM.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Newark, NJ.

Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W. (2004). A survey of recent advances in discrete input parameter discrete-event simulation optimization. *IIE Transactions*, 36(6):591–600.

Taussky, O. (1949). A recurring theorem on determinants. *Amer. Math. Monthly*, 56:672–676.

Tompkins, G. and Azadivar, P. (1995). Genetic algorithms in optimizing simulated systems. In *Proceedings of the 1995 Winter Simulation Conference*, number 757–762.

Turlach, B. (2005). Shape constrained smoothing using smoothing splines. *Computational Statistics*, 20:81–104.

Vavak, F. and Fogarty, T. (1996). Comparison of steady state and generational genetic algorithms for use in nonstationary environments. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 192–195, Piscataway, NJ. IEEE Press.

Wagner, H. M. (1959). Linear programming techniques for regression analysis. *Journal of the American Statistical Association*, 54(285):206–212.

White Jnr, K. P. (1997). An effective truncation heuristic for bias reduction in simulation output. *Simulation*, 69(6):323–334.

White Jnr, K. P., Cobb, M. J., and Spratt, S. C. (2000). A comparison of five steady-state truncation heuristics for simulation. In *Proceedings of the 2000 Winter Simulation Conference*, pages 755–760.

Whitney, J. E., I., Solomon, L. I., and Hill, S. D. (2001). Constrained optimization over discrete sets via spsa with application to non-separable resource allocation. In *Proceedings of the 2001 Winter Simulation Conference*, volume 1, pages 313–317.

Williams, D. (1991). *Probability with Martingales*. Cambridge University Press, Cambridge, UK.

Wilson, J. R. and Pritsker, A. A. B. (1978). Evaluation of startup policies in simulation experiments. *Simulation*, 31(3):79–89.

Wu, C. F. (1982). Some algorithms for concave and isotonic regression. *TIMS Studies in Management Science*, 19:105–116.

Yan, D. and Mukai, H. (1992). Stochastic discrete optimization. *SIAM Journal on Control and Optimization*, 30:594–612.

Yücesan, E. (1993). Randomization tests for initialization bias in simulation output. *Naval Research Logistics*, 40:643–663.

Zangwill, W. I. (1969). *Nonlinear Programming: A United Approach*. Prentice–Hall, Inc., Englewood Cliffs, New Jersey.

Zheng, Y.-S. (1992). On properties of stochastic inventory systems. *Management Science*, 38(1):87–103.