NORTHWESTERN UNIVERSITY

Quasi-Monte Carlo Methods for Stochastic Programming

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Industrial Engineering and Management Sciences

By

Shane Sebastian Drew

EVANSTON, ILLINOIS

December 2007

ABSTRACT

Quasi-Monte Carlo Methods for Stochastic Programming

Shane Sebastian Drew

In this thesis we discuss the issue of solving stochastic optimization problems using sampling methods. Numerical results have shown that using variance reduction techniques from statistics can result in significant improvements over Monte Carlo sampling in terms of the number of samples needed for convergence of the optimal objective value and optimal solution to a stochastic optimization problem . Among these techniques are stratified sampling and randomized Quasi-Monte Carlo sampling.

This thesis is split into three main sections:

- The first section discusses deviation probabilities for Latin Hypercube sampling (LHS), which is a type of stratified sampling. A deviation probability is the probability that the sample average differs from the true expectation by more than some quantity δ. For Monte Carlo sampling, it is known that the deviation probability approaches zero exponentially fast as the sample size grows to infinity. We show that under certain conditions, that rate of convergence is even faster for Latin Hypercube sampling.

- The second section deals with a padded sampling scheme. For problems in high dimension, it may be computationally inefficient to calculate Quasi-Monte Carlo point sets in the full dimension. Rather, we can identify which dimensions are most important to the convergence and implement a sampling scheme where only those important dimensions are sampled via Quasi-Monte Carlo sampling and the remaining dimensions are "padded" with some other type of sampling. We show that a padded sampling scheme

where the padded variables are sampled with Latin Hypercube sampling (PLHS) satisfies a normal central limit theorem.

- In the third section, we incorporate our padded sampling scheme (PLHS) into an algorithm to solve two-stage stochastic optimization problems and show some numerical results. Additionally we show that when padded sampling incorporated into single replication and two-replication stopping procedures for our algorithm, the algorithm will stop after a finite number of samples.

ACKNOWLEDGMENTS

Completing this dissertation has involved a long and interesting journey — one which I could not have made without the help and support of many people along the way.

I would like to thank first my advisor Tito Homem-de-Mello for all of his guidance and support. There were many instances over the past few years when I questioned whether this dissertation would ever become a reality, but Tito was always there to bring me back on course. I can truly say that without his persistence and understanding, this document would never have been completed (or probably even started) and I would not be where I am today. For that I am eternally grateful.

I would also like to thank the other members of my dissertation committee — Sanjay Mehrotra, Barry Nelson, and Jeff Linderoth — for their input on this thesis. Taking classes from Sanjay and Barry, along with Robert Fourer and John Birge, made me interested in optimization and simulation and gave me much of the groundwork to complete the technical work in this thesis. Jeff, along with Stephen Wright, provided me access to and technical support for the code for `sutil` and `ATR`, which greatly helped in obtaining the numerical results for this thesis.

Next, I would like to thank all of my friends and former officemates from room C229 — Peter, Zhen, Gigi, Fang, Taylan, Min, Leyla, Louis, Suzanne, Jisheng, Jeff, Jie, and Michael — my giant, sometimes dysfunctional, grad school family. I greatly enjoyed all of our conversations, both academic and non-academic, over the years. I learned so much from all of you and look forward to keeping in touch. Thank you for always being there and making my grad school experience much more fun!

Of course, I would also like to thank my real family — especially my father Richard, mother Janice, and sister Whitney — for their love and support and for helping me become

the person I am today. They never once questioned why I would leave a nice lucrative actuarial job to move halfway across the country to study something they had never even heard of. Someday, I will successfully explain to them what this dissertation is about ◡̈. (My mother's exact quote about this dissertation was "I don't even know what any of the words in the title mean!"). Additionally, I would like to mention two special people who passed away during the course of this dissertation work but would have been very proud of its result — my grandmother Gloria Costa (who would call me every couple of weeks to make sure that I was eating correctly) and my grandfather Richard Drew (who when I would visit would pull me aside to tell/ask me "I can't wait to have a Dr. Drew in the family. When are you finishing that damn paper?").

Finally, I would like to thank my fiance Kate who has been my rock for the past $3\frac{1}{2}$ years. She has always been there to celebrate with me on the good days, to cheer me up on the bad days, and to make all the in-between days much nicer too. I'm not sure how she puts up with all of my many quirks, but I am very glad that she does. Her unconditional love and support have meant the world to me!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We live in a stochastic world. Random quantities and events such as customer demand, prices of financial securities, machine failures, and the weather need to be accounted for to help us make more informed decisions. One decision-making technique is optimization where the decision-maker selects the alternative(s) that maximizes or minimizes some objective function. The notion of optimization under uncertainty, or stochastic optimization, was first developed in the 1950's (Dantzig 1955, Beale 1955, Charnes and Cooper 1959, Robbins and Monro 1951b). However, progress in the field was then relatively quiet until the 1980's when it finally became feasible to solve much larger optimization problems due to the proliferation of computing power.

In this thesis, we consider stochastic optimization problems of the form:

$$\min_{x \in X} \{g(x) := \mathbb{E}[G(x, \xi)]\} \tag{1.1}$$

where X is a subset of $\mathbb{R}^n$, ξ is a random vector in $\mathbb{R}^s$, and $G : \mathbb{R}^n \times \mathbb{R}^s \mapsto \mathbb{R}$ is a real valued function. We shall refer to (1.1) as the true optimization problem. We also assume that this problem has a finite solution.

Often though, $G(x, \xi)$ cannot be written in closed form or it cannot be easily calculated due to a large sample space. In these cases, we can approximate the expectation with a sample average:

$$\hat{g}_N(x) := \frac{1}{N} \sum_{j=1}^{n} G(x, \xi^j), \tag{1.2}$$

where the ξ^j are random samples from the distribution of ξ.

From our family of estimators $\{\hat{g}_N(\cdot)\}$, we can construct another stochastic program

$$\min_{x \in X} \hat{g}_N(x), \tag{1.3}$$

which we shall refer to as the sampled optimization problem.

Let, $\hat{v}_N$ denote the optimal objective value to the sampled optimization problem and $\hat{x}_N$ an optimal solution. Then $\hat{v}_N$ and $\hat{x}_N$ are approximations to the true optimal objective value v^* and some true optimal solution x^*.

When the samples $\xi^1, \ldots, \xi^N$ are independent and identically distributed (i.i.d.), $\hat{g}_N(x)$ is referred to as a Monte Carlo estimator of $g(x)$ and the approach of solving the sampled optimization problem is usually referred to as the *sample average approximation method* or *sample path optimization*. Approaches using Monte Carlo methods have been well-studied in the context of stochastic optimization. If x^* is the unique optimal solution to the true optimization problem, then $\hat{x}_N \to x^*$ and $\hat{v}_N \to v^*$ under some general conditions (See, e.g. Dupačová and Wets 1988, King and Rockafellar 1993, Robinson 1996, Shapiro 1991, Shapiro 1993). Another result of interest is the rate of convergence. Under suitable conditions, for some fixed $\varepsilon > 0$, $\mathbb{P}(|g(\hat{x}_N) - g(x^*)| \leq \varepsilon)$ and $\mathbb{P}(||\hat{x}_N - x^*|| \leq \varepsilon)$ both converge to one exponentially fast as the sample size N tends to infinity (Dai, Chen, and Birge 2000, Kaniovski, King, and Wets 1995). Under further conditions, $\mathbb{P}(\hat{x}_N = x^*)$ converges to one

exponentially fast as the sample size N tends to infinity (Shapiro and Homem-de-Mello (2000)). Finally, Shapiro (1991) has shown that the sequence of optimal objective values $\{\hat{v}_N\}$ satisfies a central limit theorem. Namely,

$$\sqrt{N}(\hat{v}_N - v^*) \Rightarrow \text{Normal}(0, \sigma_*^2)$$

where " $\Rightarrow$ " denotes convergence in distribution and $\sigma_*^2 := Var[G(x^*)]$. Thus the convergence of optimal objective values is of order $\frac{1}{\sqrt{N}}$.

Often though, either the sample size N required to guarantee a small error is extremely large or it is computationally expensive to evaluate the function $G(x, \xi)$ at fixed values of ξ. A natural next step is to consider variance reduction techniques from the simulation and statistics literature and apply them to the stochastic optimization problem. This research is still in the relatively early stages (Bailey, Jensen, and Morton 1999, Dantzig and Glynn 1990, Emsermann and Simon 2000, Higle 1998, Infanger 1994, Shapiro and Homem-de-Mello 1998). Among these variance reduction techniques are antithetic variates, control variates, and importance sampling (see Law and Kelton (2000)).

We focus on two other variance reduction techniques: stratified sampling and randomized Quasi-Monte Carlo sampling. The main idea of stratified sampling is to split the sample space into strata and to sample within each stratum roughly proportionally to that stratum's probability. This is done to reduce the likelihood that the random samples are clustered in one general area of the sample space. In Quasi-Monte Carlo sampling (QMC), we take the idea of stratification one step further. Rather than randomly sampling points within each stratum, we choose them deterministically while obeying some constraints regarding the distance between any two sample points. Since the variance of a deterministic sampling scheme is identically zero, some uniform shuffling scheme may be employed on the deterministic QMC points to make them random. Such methods are called Randomized Quasi-Monte

Carlo (RQMC) methods. The goal of QMC and RQMC is for the point set to have a uniform dispersion across the sample space. The primary issue however with these sampling schemes is that the samples are no longer independent or identically distributed and thus many of the results from classical probability no longer hold.

There are a wide variety of Quasi-Monte Carlo methods (which we will discuss in Chapter 2). Some of these methods have been shown to have errors on the order of $\frac{(\log N)^s}{N}$ when computing the sample average, which is asymptotically superior to the error for Monte Carlo sampling. Quasi-Monte Carlo sampling does have its drawbacks though. For one, it can be computationally expensive to generate QMC point sets for high dimensions. Also, while the $\frac{(\log N)^s}{N}$ rate of convergence is asymptotically superior to the Monte Carlo rate of $\frac{1}{\sqrt{N}}$, it is dependent on the dimension and often does not become superior to Monte Carlo until N is extremely large. This is impractical for a sampling algorithm. However, QMC in practice often outperforms its theoretical rate of convergence. This suggests that the sample spaces of these problems are really in some lower dimension— i.e., that some random variables are considerably more important to the problem than others. This leads to the notion of hybrid or padded sampling schemes where the most important variables are sampled using Quasi-Monte Carlo and the remaining variables are sampled using a computationally less expensive scheme such as Monte Carlo or some form of stratified sampling. We will specifically look at padding with Monte Carlo sampling (PMC) and padding with Latin Hypercube sampling (PLHS).

This dissertation will cover three projects. First, we look at deviation probabilities of sample averages from Latin Hypercube sampling. While large deviations results for i.i.d. sampling have been well-studied (see Section 3.2.1) and general formulas for non-i.i.d. sampling have been derived, there were previously no specific results for deviations probabilities under Latin Hypercube sampling. We will examine these probabilities and show that under

certain assumptions the rate of convergence of the deviation probability of a sample average (not in the context of stochastic optimization) is indeed faster for Latin Hypercube than the exponential rate of convergence from standard Monte Carlo sampling. Homem-de Mello (2006) has since extended these results to optimal objective values of stochastic optimization problems.

Second, we examine a padded sampling scheme where the padded variables are sampled with Latin Hypercube sampling (PLHS). Ökten, Tuffin, and Burago (2006) have already shown that a padded sampling scheme where the padded variables are sampled with Monte Carlo sampling (PMC) satisfies a normal central limit theorem for bounded functions and that the asymptotic variance of this distribution is no worse than the asymptotic variance from pure Monte Carlo sampling. Owen (1992) and Stein (1987) have shown the same results for pure LHS versus Monte Carlo sampling. We will show that a central limit theorem for bounded functions also holds for PLHS and that its asymptotic variance is no worse than Monte Carlo, LHS, or PMC.

Third, we implement PLHS and PMC into sampling algorithms to solve stochastic program (1.3). To date, we have found no other papers using padded sampling schemes to solve stochastic programs. We also incorporate padded sampling into the stopping criteria for these algorithms and show that these stopping criteria are asymptotically valid. The lower variance provided by padded sampling schemes should provide tighter confidence intervals in the stopping criteria than their Monte Carlo counterparts.

The structure of this dissertation is as follows. In Chapter 2 we give an overview of sampling methods with a focus on Latin Hypercube sampling (a type of stratified sampling proposed by McKay, Beckman, and Conover (1979)) and Quasi-Monte Carlo sampling. In Chapter 3 we show our large deviations results for Latin Hypercube sampling. In Chapter 4, we show that a padded sampling scheme where the padded variables are sampled with

Latin Hypercube sampling (PLHS) satisfies a normal central limit theorem provided that the function of interest is bounded (again not in the context of stochastic optimization). In Chapter 5 we present an algorithm which uses padded sampling to solve two-stage stochastic linear programs and illustrate some numerical results. We also show that the use of padded sampling in stopping criteria for stochastic programs yields asymptotically valid confidence intervals before offering some final remarks and comments in Chapter 6.

Chapter 2

An Overview of Sampling Methods

2.1 Introduction

Suppose that X is a random variable in $\mathbb{R}^s$ with all of its arguments mutually independent, $g : \mathbb{R}^s \mapsto \mathbb{R}$ is a measurable function, and we wish to estimate $I = \mathbb{E}[g(X)]$. One way to do this is via numerical integration. We choose a point set $\{\xi^1, \ldots, \xi^N\}$ from the sample space and then calculate the sample average $\hat{I} = \frac{1}{N}\sum_{j=1}^{N} g(\xi^j)$ from that point set. When the elements of the point set are independent and identically distributed (i.e., each point is sampled randomly from the entire sample space), the numerical integration method is called a Monte Carlo method. It is well known from statistics that if the function $g(X)$ has a finite second moment then the error of the sample average approximation $\hat{I}_{MC}$ is of order $\frac{1}{\sqrt{N}}$. However, with Monte Carlo sampling, there is always the chance that the random samples are clustered in one region of the sample space. One way to rectify this is stratified sampling — where the sample space is split into strata and the point set contains a specified number of random samples from each strata. We will concentrate specifically on one type of stratified sampling called Latin Hypercube sampling. A second method that deals with the clustering problem is Quasi-Monte Carlo sampling, where the point sets are entirely deterministic.

2.2 Latin Hypercube Sampling

Latin Hypercube sampling was first proposed by McKay et al. (1979). The algorithm works as follows when the random variable X is uniformly distributed on $[0,1]^s$ and the dimensions $j = 1, \ldots, s$ are all mutually independent of each other:

Algorithm LHS: For each dimension $j = 1, \ldots, s$:

1. Generate

$$Y_j^1 \sim U\left(0, \frac{1}{N}\right), Y_j^2 \sim U\left(\frac{1}{N}, \frac{2}{N}\right), \ldots, Y_j^N \sim U\left(\frac{N-1}{N}, 1\right);$$

2. Let $\xi_j^i := Y_j^{\pi(i)}$ where π is a random permutation of $1, \ldots, N$. ■

The i^{th} sampled point is then just the s-tuple $\xi^i = (\xi_1^i, \ldots, \xi_s^i)$. The result is that each stratum from each dimension is sampled exactly once. See Figure 2.1 for the case where dimension $s = 2$ and sample size $N = 4$. In the case that the arguments are not of uniform distribution, the algorithm can be modified by drawing the samples from the uniform distribution and then applying the inverse transform method to obtain the desired distribution. We then can evaluate $g(\cdot)$ at the sampled points and calculate the sample average. If we altered step 1 of *Algorithm LHS* so that $Y_j^i = \frac{i-0.5}{N}$ (the midpoint of the interval rather than a random point), then the sampling scheme is called Centered Latin Hypercube sampling (CLHS).

McKay et al. (1979) show that both Latin Hypercube sampling and Centered Latin Hypercube sampling give an unbiased estimate of the mean. Further, if the function g is monotone in each argument when all of the other arguments are held fixed, then the sample variance of the estimator under Latin Hypercube sampling is no worse than the sample variance under Monte Carlo sampling. Hoshino and Takemura (2000) extend this to the

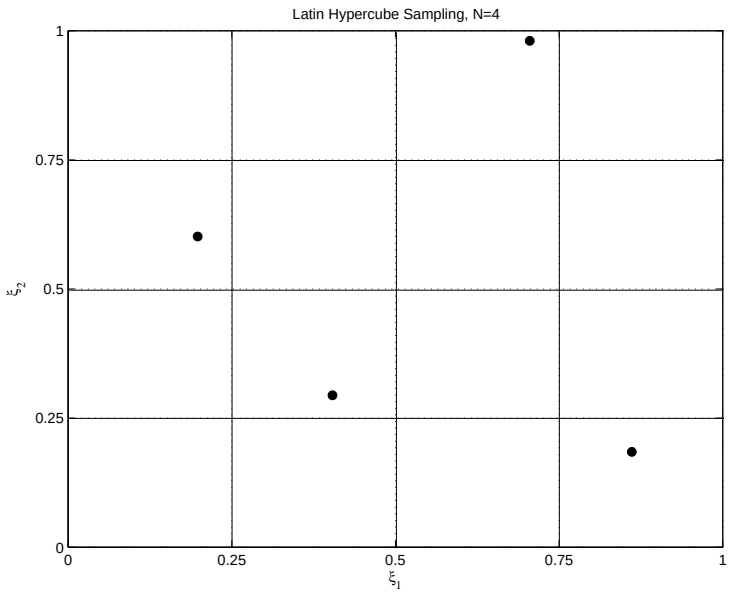

Figure 2.1: Latin Hypercube Sampling, $N = 4, s = 2$

case where $g(\cdot)$ is monotone in all but one of its arguments. Owen (1997b) generalizes the relationship between Monte Carlo and LHS variance for all measurable functions and shows that $Var_{LHS} \leq \frac{N}{N-1} Var_{MC}$, where Var_{LHS} and Var_{MC} are the variances under Latin Hypercube and crude Monte Carlo sampling respectively. Thus the variance under Latin Hypercube sampling is no worse than Monte Carlo sampling asymptotically, but could be for general N.

Stein (1987) splits the function $g(\cdot)$ into a constant component μ plus first order terms plus a residual component (this is called the ANOVA decomposition of g), i.e.,

$$g(X) = \mu + g_1(X^1) + \cdots + g_s(X^s) + g_{resid}(X). \tag{2.1}$$

Stein (1987) also shows that, asymptotically, the sample variance from Latin Hypercube sampling is equal to the variance of the residual term in the ANOVA decomposition. As a result of this Latin Hypercube sampling performs particularly well on functions that are

separable or nearly separable. Loh (1996) extends this result to the multivariate case where $g : \mathbb{R}^s \mapsto \mathbb{R}^m$. Finally, Owen (1992) shows that Latin Hypercube sampling satisfies a Central Limit Theorem with the variance equal to the variance of the residual term.

2.3 Quasi-Monte Carlo Sampling

One of the problems with Monte Carlo sampling or any other random sampling scheme is that it only gives a probabilistic error bound, i.e., on average a sample of size N will give an error of order $\frac{1}{\sqrt{N}}$ for any function. One could instead deterministically choose a point set of size N that outperforms the average error for most functions. Such methods are called Quasi-Monte Carlo (QMC) methods. Comprehensive reviews of Quasi-Monte Carlo methods can be found in Niederreiter (1992) and L'Ecuyer and Lemieux (2002).

In Quasi-Monte Carlo sampling, we wish to select points $\xi^1, \ldots, \xi^N$ that are approximately uniformly spaced avoiding large gaps or clusters. More formally, if the sample space is Ω, then for any measurable subset J of Ω, we would like

$$\frac{\sum_{j=1}^{N} \chi_J(\xi^j)}{N} \approx \frac{\lambda_s(J)}{\lambda_s(\Omega)}$$

where $\lambda_s(\cdot)$ is the s-dimensional Lebesgue measure and $\chi_J(\cdot)$ is the characteristic function of J (equal to 1 if $\xi^j \in J$ and 0 if not). Without loss of generality we will assume that $\Omega = [0,1)^s$ (which has Lebesgue measure one) as we can use the inverse transform on uniform random variables to obtain other distributions. The difference between the empirical distribution of the Quasi-Monte Carlo point set and the uniform distribution is quantified by the *discrepancy*. There are various notions of discrepancy. We will use the star discrepancy

which is defined as

$$D_N^*(\xi^1, \ldots, \xi^N) = \sup_{J \in \mathfrak{J}^*} \left| \frac{\sum_{j=1}^N \chi_J(\xi^j)}{N} - \lambda_s(J) \right|, \tag{2.2}$$

where $\mathfrak{J}^*$ is the family of all subintervals of $[0,1)^s$ of the form $\prod_{i=1}^s [0, u_i)$. The star-discrepancy will always be between zero and one.

The Koksma-Hlawka inequality shows that the error from the estimate $\hat{I}_{QMC} = \frac{1}{N} \sum_{j=1}^N g(\xi^j)$ is bounded above by the product of the star-discrepancy and the total Hardy-Krause variation (Hardy, 1905) of the function $g(\cdot)$ (denoted $V(g)$):

$$|\hat{I}_{QMC} - I| \leq D_N^*(\xi^1, \ldots, \xi^N) V(g). \tag{2.3}$$

As a result, most of the research on Quasi-Monte Carlo methods has focused on finding low-discrepancy sequences.

One of the first Quasi-Monte Carlo sequences was the one-dimensional van der Corput sequence which is based upon the radical inverse sequence. Let $b \geq 2$ be an arbitrary integer called the base. Every integer $i \geq 0$ has a unique digit expansion in base b:

$$i = \sum_{j=0}^{\infty} d_j(i) b^j. \tag{2.4}$$

The radical inverse function is

$$\phi_b(i) = \sum_{j=0}^{\infty} \frac{d_j(i)}{b^{j+1}} \tag{2.5}$$

and the van der Corput sequence in base b is then the sequence $\xi^0, \xi^1, \xi^2, \ldots$ with $\xi^i = \phi_b(i)$. In base 2 this is $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \ldots$. The discrepancy of the van der Corput sequence is $O\left(\frac{\log(N)}{N}\right)$.

The Halton sequence extends the van der Corput sequence to multiple dimensions by

using base b_j for dimension j where b_j is the j^{th} smallest prime number (with $b_1 = 2$). Thus the Halton sequence in s dimensions is $\xi^0, \xi^1, \xi^2, \ldots$ with $\xi^i = (\phi_{b_1}(i), \ldots, \phi_{b_s}(i))$. The first N terms of the Halton sequence has discrepancy $O\left(\frac{(\log(N))^s}{N}\right)$.

If we know the number of sample points N in advance, we can improve the discrepancy of the Halton sequence by using a Hammersley point set (Hammersley, 1960). In a Hammersley point set of dimension s, the first dimension consists of equally spaced points while the remaining $s-1$ dimensions are a Halton sequence of length N where the bases are the smallest $s-1$ prime numbers:

$$\xi^i = \left(\frac{i}{N}, \phi_{b_1}(i), \ldots, \phi_{b_{s-1}}(i)\right), \quad i = 1 \ldots N$$

The discrepancy of a Hammersley point set is $O\left(\frac{(\log(N))^{s-1}}{N}\right)$.

There are two problems with the sequences and point sets above. First, we would like to use the same base for each dimension. Second, as the dimension s goes to infinity, the coefficient on the leading term of the discrepancy grows superexponentially. We would like that coefficient to go to zero. Both of these problems are solved with a (t, s)-sequence.

Before we define a (t, s)-sequence, we need a couple of definitions. An elementary interval of base b in dimension s is a subinterval E of the form

$$E = \prod_{j=1}^{s} \left[\frac{a_j}{b^{d_j}}, \frac{a_j + 1}{b^{d_j}}\right)$$

for nonnegative integers $\{a_j\}$ and $\{d_j\}$ with $a_j < b^{d_j}$ for all j. E then has volume $b^{-\sum_j d_j}$.

Let m be a nonnegative integer. Then a finite sequence of $N = b^m$ points is called a $(0, m, s)$-net in base b if every elementary interval in base b of volume $\frac{1}{b^m}$ contains exactly one point of the sequence. Latin Hypercube sampling is a special case of this with each realization being a $(0, 1, s)$-net in base $b = N$.

This is easily extended to any integer t with $0 \leq t \leq m$. A finite sequence of $N = b^m$ points is called a (t, m, s)-net in base b if every elementary interval in base b of volume $\frac{1}{b^{m-t}}$ contains exactly b^t points of the sequence. Note, that any (t, m, s)-net is also a (u, m, s)-net for any integer $u \in [t, m]$. Thus smaller values of t are more desirable. The discrepancy of a (t, m, s)-net in base b is $O\left(\frac{(\log(N))^{s-1}}{N}\right)$ – the same as a Hammersley point set except that the coefficient of the leading term now goes to zero as the number of sample points N goes to infinity.

A digital (t, m, s)-net in base $b \geq 2$ (prime) is a (t, m, s)-net generated by digit expansion. Let $d(i) = [d_1(i) \quad d_2(i) \quad \cdots]^T$ be defined from the base b expansion of positive integer i in E quation (2.4) and let $C_1, \ldots C_s$ be $(m \times m)$-matrices over the finite field $\mathfrak{F}_b$. For each sample point $i = 1, \ldots, N$ and each dimension $j = 1, \ldots, s$, calculate $[y_{j1}(i) \quad \ldots \quad y_{j\infty}(i)]^T = C_j d(i)^T$. Then the point set is comprised of $\xi^1, \ldots, \xi^N$ where

$$\xi^i = \left(\sum_{k=1}^{\infty} \frac{y_{1k}(i)}{b^k}, \ldots, \sum_{k=1}^{\infty} \frac{y_{sk}(i)}{b^k} \right).$$

A sequence of points $\xi^0, \xi^1, \xi^2, \ldots$ is called a (t, s)-sequence in base b if for all integers $k \geq 0$ and $m > t$, the set of ξ^i with $kb^m \leq i \leq (k+1)b^m$ is a (t, m, s)-net in base b. One special case is the van der Corput sequence which is just a $(0, 1)$-sequence in base b. Like the (t, m, s)-net, any (t, s)-sequence in base b is also a (u, s) sequence in base b for all integers $u \geq t$. The first $N \geq 2$ terms of a (t, s)-sequence in base b has discrepancy $O\left(\frac{(\log(N))^s}{N}\right)$.

While these Quasi-Monte Carlo methods compare favorably to Monte Carlo sampling methods, there are still two major disadvantages:

- It is often difficult to compute the quantities in the upper bound of the Koksma-Hlawka inequality. Also, we do not know how tight that bound is. Thus computing an accurate error bound for Quasi-Monte Carlo integration can be problematic.

- While the Quasi-Monte Carlo error bound of $\frac{(\log(N))^s}{N}$ is asymptotically superior to the Monte Carlo error bound of $\frac{1}{\sqrt{N}}$, it is not advantageous until N is very large, unless the dimension s is small. Even for $s = 5$, we must have $N \geq 3.4 \times 10^{15}$ for QMC to be theoretically advantageous.

We usually deal with the first problem by employing randomization into the Quasi-Monte Carlo point set. We remedy the latter by looking at the effective dimension of the sample space rather than the given dimension.

2.3.1 Randomized Quasi-Monte Carlo

In the previously discussed Quasi-Monte Carlo methods, the point set is entirely deterministic and thus the estimator $\hat{I} = \frac{1}{N}\sum_{j=1}^{N} g(\xi^j)$ has zero variance. This makes it difficult to calculate the integration error $|\hat{I}_{QMC} - I|$. In a Randomized Quasi-Monte Carlo method, each individual sample point is now uniformly distributed over the sample space, but the point set as a whole still preserves the low-discrepancy property. Randomized QMC methods are discussed in detail in Fox (2000) and Owen (2000). Since the point set is now random, errors can be estimated using standard methods such as multiple independent replications.

Cranley and Patterson (1976) proposed a rotation modulo one for a QMC method called an integration lattice. If the original Quasi-Monte Carlo sequence is $\{a^i\}$, then the terms of the randomized QMC sequence are

$$\xi^i = (a^i + U) \mod 1$$

where U is uniformly distributed on $[0, 1)^s$ and the addition and modulo are performed componentwise. Tuffin (1996) applied such rotations to digital nets but found that while the sample points exhibit properties of uniformity, they are no longer nets.

Owen (1995) proposed the scrambled digital net. To begin, each dimension is split into b elementary intervals of the form $\left[\frac{k}{b}, \frac{k+1}{b}\right)$. Then one dimension at a time, random permutations are generated to reorder the elementary intervals. In the second pass, the b intervals in each dimension are each subdivided into b subintervals. For each interval, a random permutation is generated to reorder the subintervals. This process continues.

More formally, let $\{A^i\}$ be a (t, m, s)-net or (t, s)-sequence in base b with $A^i = (A^i_1, \ldots, A^i_s)$. Then each A^i_j can be expanded as a radical inverse sequence in base b:

$$A^i_j = \sum_{k=1}^{\infty} \frac{d_{jk}(i)}{b^k}$$

where $0 \leq d_{jk}(i) < b$ for all i, j, k. To scramble the net, we generate independent permutations $\pi_{j\cdot}(\cdot)$ of $\{0, \ldots . b-1\}$ for each digit k given the values of the first $k-1$ digits:

$$\begin{aligned}
\xi^i_{j1} &= \pi_j(d_{j1}(i)) \\
\xi^i_{j2} &= \pi_{jd_{j1}(i)}(d_{j2}(i)) \\
\xi^i_{j3} &= \pi_{jd_{j1}(i)d_{j2}(i)}(d_{j3}(i)) \\
&\vdots \\
\xi^i_{jk} &= \pi_{jd_{j1}(i)d_{j2}(i)\cdots d_{jk-1}(i)}(d_{jk}(i)).
\end{aligned}$$

The randomized sequence then comes from computing the radical inverse function using the permuted digits. The i^{th} term is $\xi^i = (\xi^i_1, \ldots, \xi^i_s)$ with

$$\xi^i_j = \sum_{k=1}^{\infty} \frac{\xi^i_{jk}}{b^k}.$$

One nice property of scrambled nets is that any scrambled (t, m, s)-net in base b is also a (t, m, s)-net with probability one (similarly any scrambled (t, s)-sequence in base b is also

a (t, s)-sequence in base b with probability one) (Owen 1995). Thus all of the properties of the (t, m, s)-nets and (t, s)-sequences can also be applied to the scrambled nets and sequences. Another important property is that the point set from a scrambled net is uniformly distributed over the unit hypercube. This is a consequence of the scrambling and is independent of whether or not the pre-scrambled point set was a (t, m, s)-net. As a result, the integration error can be calculated by taking multiple independent replications of the scrambled net (from the same original net). While in general, the integration error from a (t, m, s)-net is $O\left(\frac{(\log N)^{s-1}}{N}\right)$, Owen (1997b) shows that for smooth functions, scrambled (t, m, s)-nets give an integration error of $O\left(\frac{(\log N)^{\frac{s-1}{2}}}{N^{\frac{3}{2}}}\right)$, which is an improvement on both the Monte Carlo and general Quasi-Monte Carlo integration errors.

2.3.2 Effective Dimension

A second problem with Quasi-Monte Carlo sampling is that the integration error is dependent on the number of dimensions of the sample space while Monte Carlo sampling is independent of the dimension. So while asymptotically the Quasi-Monte Carlo rate is superior to Monte Carlo, for problems in large dimension it does not theoretically become so until the number of sample points is extremely large. Yet for many problems, numerical results show that Quasi-Monte Carlo methods have a much lower integration error, even for reasonable sample sizes. This suggests that either the upper bound from the Koksma-Hlawka inequality is not tight or that the problem is in some lower dimension.

If the function $g(\cdot)$ is square-integrable, then $g(\cdot)$ can be written

$$g(\xi) = \sum_{A \subseteq \{1,\ldots,s\}} g_A(\xi) \tag{2.6}$$

where $g_A(\xi)$ depends only on the components ξ_j with $j \in A$. Also, $g_A(\xi)$ satisfies the

properties:

1. $\int_0^1 g_A(\xi)d\xi_j = \begin{cases} 0, & \text{if } j \in A \\ g_A(\xi), & \text{if } j \notin A. \end{cases}$
2. $\int_{[0,1)^s} g_u(\xi)g_v(\xi)d\xi = 0, \quad \text{if } u \neq v.$

For each subset A, $g_A(\xi)$ is determined by looking at the portion of g that is not determined by subsets of A and then averaging over all of the components not in A

$$g_A(x) = \int_{[0,1)^{A^c}} \left(g(\xi) - \sum_{v \subset A} g_v(\xi) \right) d\xi^{A^c}. \tag{2.7}$$

The goal of any sampling method is to estimate $\bar{g} := \int_{[0,1)^s} g(\xi)d\xi$ (equivalent to letting A be the empty set in (2.7)). From the properties above, the variance of g, defined as $\sigma^2 := \int (g(\xi) - \bar{g})^2 d\xi$, satisfies

$$\sigma^2 = \sum_{A \subseteq \{1,\ldots,s\}} \sigma_A^2 \tag{2.8}$$

where $\sigma_A^2 := \int g_A(\xi)^2 d\xi$. Hence, this is called the ANOVA (analysis of variance) decomposition of g. One can then define the effective dimension of the problem based on the terms of the ANOVA decomposition that contribute most toward the overall variance. Caflisch, Morokoff, and Owen (1997) define two notions of effective dimension: the truncation dimension and the superposition dimension.

The effective dimension of g in the truncation sense is the smallest integer s_T such that

$$\sum_{A \subset \{1,2,\ldots,s_T\}} \sigma_A^2 \geq (1 - \varepsilon)\sigma^2. \tag{2.9}$$

Typically, ε is chosen to be 0.01. Note that the value of s_T is dependent on the order in which the input variables are indexed. This definition implies that only a small number

of the input variables are important to the problem, thus there is little need for variance reduction techniques on the remaining variables. This leads to the concept of padding where the important variables are integrated using a QMC or randomized QMC point set and the remaining variables are integrated using something computationally less expensive such as the midpoint of the interval, a Monte Carlo sample (Spanier 1995), or a Latin Hypercube sample (Owen 1998). Example of this technique will be described in Chapters 4 and 5.

The effective dimension of g in the superposition sense is the smallest integer s_S such that

$$\sum_{|A|\leq s_S} \sigma_A^2 \geq (1-\varepsilon)\sigma^2. \tag{2.10}$$

It can be shown that $0 \leq s_T \leq s_S \leq s$. One way to interpret the effective dimension in the superposition sense is that $g(\xi)$ can be replaced by a function of dimension s_S which would explain $(1-\varepsilon)\%$ of its variance. Thus the Quasi-Monte Carlo integration error should depend on something closer to the effective dimension rather than the true dimension. Owen (2002) has shown that low effective dimension in the superposition sense is necessary for scrambled (0,m,s)-nets to beat Monte Carlo methods by a wide margin for high dimensions and modest sample sizes.

Owen (2003) proposes the concept of dimension distribution in the superposition sense. He defines a random variable D with range $\{1,\ldots,s\}$ such that $\mathbb{P}(D=k) = \frac{1}{\sigma^2}\sum_{A:|A|=k}\sigma_A^2$ (i.e., the percentage of the variance attributed to terms of order k). Since it is difficult to compute the full distribution of D, Liu and Owen (2003) propose calculating moments of D, which turns out to be a far simpler task. For example, the mean of D is $\mathbb{E}[D] = \frac{1}{\sigma^2}\sum_{j=1}^{s}\bar{\tau}_{\{j\}}$ where $\bar{\tau}_{\{j\}}$ is given by

$$\bar{\tau}_{\{j\}} = \frac{1}{2}\mathbb{E}\left[(g(U_1,\ldots,U_s) - g(U_1,\ldots,U_{j-1},V_j,U_{j+1},\ldots,U_s))^2\right] \tag{2.11}$$

and $(U_1, \ldots, U_s)$ and $(V_1, \ldots, V_s)$ are random sequences such that each pair $\{U_j, V_j\}$ is i.i.d. Higher moments of D can be computed accordingly.

Chapter 3

Large Deviations for Latin Hypercube Sampling

3.1 Introduction

Suppose we wish to calculate $\mathbb{E}[g(X)]$ where $X = [X^1, \ldots, X^d]$ is a random vector in $\mathbb{R}^d$ and $g(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$ is a measurable function. Further, suppose that the expected value is finite and cannot be written in closed form or be easily calculated, but that $g(X)$ can be easily computed for a given value of X. Let $\mathbb{E}[g(X)] = \mu \in (-\infty, \infty)$. To estimate the expected value, we can use the sample average approximation:

$$\frac{1}{n} S_n = \frac{1}{n} \sum_{i=1}^{n} g(X_i(\omega)) \tag{3.1}$$

where the $X_i(\omega)$ are random samples of X. When the $X_i(\omega)$ are i.i.d. (i.e., Monte Carlo sampling), by the law of large numbers the sample average approximation should approach the true mean μ (with probability one) as the number of samples n becomes large. Large deviations theory ensures that the probability that the sample average approximation devi-

ates from μ by a fixed amount $\delta > 0$ approaches zero exponentially fast as n goes to infinity. Formally, this is expressed as

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\left|\frac{1}{n}S_n - \mu\right| > \delta\right) \;=\; -\beta_\delta,$$

where β_δ is a positive constant.

The description above, of course, is a small fraction of a much more general theory, but conveys a basic concept — that one obtains *exponential convergence* of estimators under certain conditions. This idea has found applications in numerous areas, from simulation to telecommunications to physics; we refer to classical books in the area such as Dembo and Zeitouni (1998) and Bucklew (2004) for further discussions.

Despite the exponential convergence results mentioned above, it is well known that Monte Carlo methods have some drawbacks, particularly when one wants to calculate the *errors* corresponding to given estimates. Although the theory behind such calculations — notably the Central Limit Theorem — is solid, in practice the error may be large even for large sample sizes. That has led to the development of many variance reduction techniques as well as alternative sampling methods (see, e.g., Law and Kelton 2000 for a general discussion of this topic).

One alternative approach for sampling the $X_i(\omega)$ is called *Latin Hypercube sampling* (LHS, for short), introduced by McKay et al. (1979) . Broadly speaking, the method calls for splitting each dimension into n strata (yielding n^d hypercubes) and, for every dimension, sampling all n strata exactly once. This technique has been extensively used in practice, not only because of simplicity of implementation but also because of its nice properties. Indeed, McKay et al. (1979) show that if $g(X)$ is monotone in each of its arguments, then the variance of the estimator obtained with LHS (call it Var^{LHS}) is no larger than the variance from Monte Carlo sampling (Var^{MC}). Hoshino and Takemura (2000) extend this result to

the case where $g(\cdot)$ is monotone in all but one of its arguments. Stein (1987) writes the ANOVA decomposition of g, i.e.,

$$g(X) = \mu + g_1(X^1) + \cdots + g_d(X^d) + g_{resid}(X) \tag{3.2}$$

and shows that, asymptotically, the sample variance from Latin Hypercube sampling is just equal to the variance of the residual term and is no worse than the variance from Monte Carlo sampling. Loh (1996) extends this result to the multivariate case where $g : \mathbb{R}^d \mapsto \mathbb{R}^m$. Owen (1997a) shows that for any n and any function g, $\text{Var}^{LHS} \leq \frac{n}{n-1}\text{Var}^{MC}$. Also, Owen (1992) shows that LHS satisfies a Central Limit Theorem with the variance equal to the variance of the residual term. For further details on LHS, we refer the reader to Section 2.2 of this thesis.

The discussion above shows that the LHS method has been well studied and possesses many nice properties. However, to the best of our knowledge there have been no studies on the exponential convergence of estimators obtained with LHS. Thus, it is of interest to know whether large deviations results hold under Latin Hypercube sampling. This is by no means a trivial question — since the $X_i(\omega)$ are no longer i.i.d. under LHS, Cramér's Theorem (which is the basic pillar of the results for i.i.d. sampling) can no longer be applied.

In this chapter, we study the problem above. We derive conditions under which large deviations results hold under Latin Hypercube sampling. More specifically, our results apply when the integrand function is of one of the following types: one-dimensional; multi-dimensional but separable (i.e., functions with no residual term); multi-dimensional with a bounded residual term; and multi-dimensional functions that are monotone in each component. In the case of functions with a bounded residual term, our results hold provided that the deviation we are measuring is large enough. Further, in all the situations above, we show that the upper bound for the large deviations probability is lower under LHS than under

Monte Carlo sampling. Jin, Fu, and Xiong (2003) show this property holds when negatively dependent sampling is used to estimate a probability quantile of continuous distributions, whereas we prove it for the situations mentioned above.

The particular application that motivates our work arises in the context of stochastic optimization. For completeness, we briefly review the main concepts here. Consider a model of the form

$$\min_{y \in Y} \{h(y) := \mathbb{E}[H(y, X)]\}, \tag{3.3}$$

where Y is a subset of $\mathbb{R}^n$, X is a random vector in $\mathbb{R}^d$ and $H : \mathbb{R}^n \times \mathbb{R}^d \rightarrowtail \mathbb{R}$ is a real valued function. We refer to the problem above as the "true" optimization problem. Let y^* denote the optimal solution of (3.3) (assume for simplicity this solution is unique), and let ν^* denote the optimal value of (3.3).

Consider now a family $\{\hat{h}_N(\cdot)\}$ of random approximations of the function $h(\cdot)$, each $\hat{h}_N(\cdot)$ being defined as

$$\hat{h}_N(y) \;:=\; \frac{1}{N} \sum_{j=1}^{N} H(y, X_j), \tag{3.4}$$

where $X_1, \ldots, X_N$ are independent and identically distributed samples from the distribution of X. Then, one can construct the corresponding approximating program

$$\min_{y \in Y} \hat{h}_N(y). \tag{3.5}$$

An optimal solution $\hat{y}_N$ of (3.5) provides an approximation (an estimator) of the optimal solution y^* of the true problem (3.3). Similarly, the optimal value $\hat{\nu}_N$ of (3.5) provides an approximation of the optimal value ν^* of (3.3).

Many results describing the convergence properties of $\{\hat{y}_N\}$ and $\{\hat{\nu}_N\}$ exist; see, for instance, Shapiro (1991, 1993), King and Rockafellar (1993), Kaniovski et al. (1995), Robinson (1996), Dai et al. (2000). Broadly speaking, these results ensure that, under mild conditions,

$\hat{y}_N$ converges to y^* and $\hat{\nu}_N$ converges to ν^*. In case the function $H(\cdot, x)$ is *convex* and *piecewise linear* for all x — which is setting in stochastic linear programs — and the distribution of X has finite support, a stronger property holds; namely, the probability that $\hat{y}_N$ *coincides with* y^* goes to one exponentially fast, i.e.,

$$\lim_{N\to\infty} \frac{1}{N} \log \left[\mathbb{P}(\hat{y}_N \neq y^*)\right] \;=\; -\beta \tag{3.6}$$

for some $\beta > 0$ — see Shapiro and Homem-de-Mello (2000), Shapiro, Homem-de-Mello, and Kim (2002). A similar property holds in case the feasibility set Y is finite (Kleywegt, Shapiro, and Homem-de-Mello, 2001) . The importance of results of this type lies in that they allow for an estimation of a sample size that is large enough to ensure that one obtains an ε-optimal solution with a given probability; for example, in the case of finite feasibility set mentioned above, it is possible to show that if one takes

$$N \;\geq\; \frac{C}{\varepsilon^2} \log\left(\frac{|Y|}{\alpha}\right)$$

— where C is a constant that depends on the variances of the random variables $H(y, X)$ — then the probability that $|h(\hat{y}_N) - h(y^*)| < \varepsilon$ is at least $1 - \alpha$ (Kleywegt et al., 2001). Besides its practical appeal, this conclusion has implications on the *complexity* of solving stochastic optimization problems; we refer to Shapiro and Nemirovski (2004) for details.

The results described above, although very helpful, require that the estimators in (3.4) be constructed from i.i.d. samples. However, it is natural to consider what happens in case those estimators are constructed from samples generated by other methods such as LHS. Numerical experiments reported in the literature invariably show that convergence of $\{\hat{y}_N\}$ and $\{\hat{\nu}_N\}$ improves when LHS is used, but no formal results have been established. In particular, a standing question is: do exponential convergence results such as (3.6) still hold

under LHS?

We illustrate this point with a very simple example. Suppose that X is discrete uniform on $\{-2,-1,0,1,2\}$. The *median* of X (which in this case is evidently equal to zero) can be expressed as the solution of the problem $\min_{y\in\mathbb{R}} \mathbb{E}|X-y|$. Note that this is a stochastic optimization problem of the form (3.3), with $H(y,X)=|X-y|$ — which is a convex piecewise linear function. As before, let $X_1,\ldots,X_N$ be i.i.d. samples of X. Clearly, the approximating solution $\hat{y}_N$ is the median of $X_1,\ldots,X_N$. Note that, when N is odd, $\hat{y}_N$ is nonzero if and only if at least half of the sampled numbers are bigger than zero (or less than zero). That is, by defining a random variable Z with binomial distribution $B(N,2/5)$ we have

$$\mathbb{P}(Z \geq N/2) \;\leq\; \mathbb{P}\left(\hat{y}_N \neq 0\right) \;\leq\; 2\mathbb{P}(Z \geq N/2)$$

and thus

$$\frac{1}{N}\log \mathbb{P}\left(\hat{y}_N \neq 0\right) \;\approx\; -I(1/2),$$

where $I(\cdot)$ is the rate function of the Bernoulli distribution (see Section 3.2.1 for a precise definition).

Suppose now that the samples $X_1,\ldots,X_N$ are generated using LHS. It is easy to see that there are at least $\lfloor \frac{2N}{5} \rfloor$ and at most $\lfloor \frac{2N}{5} \rfloor + 1$ numbers in $\{1,2\}$ (similarly for $\{-2,-1\}$). Thus, we have that $\hat{y}_N \neq 0$ only if

$$\lfloor \frac{2N}{5} \rfloor + 1 \;\geq\; N/2.$$

Clearly, this is impossible for $N \geq 9$, so $\mathbb{P}(\hat{y}_N \neq 0) = 0$ for N large enough. That is, the asymptotic rate of convergence of $\{\hat{y}_N\}$ to the true value is *infinite*, as opposed to the exponential rate with positive constant obtained with Monte Carlo. A natural question that arises is, does such result hold in more general settings?

The main tool underlying the derivation of (3.6) is the theory of large deviations, particularly Cramér's Theorem for multi-dimensional spaces. Thus, in order to address the issue of whether (3.6) holds when the estimators are constructed using LHS, it is necessary to establish first some large deviations results for LHS. This is the goal of this paper.

The remainder of the paper is organized as follows. In Section 3.2, we give some background on large deviations theory. We also state some of the calculus results needed later in the paper. In Section 3.3, we show our results for functions in one-dimension. In Section 3.4, we extend the one-dimensional results to separable functions with multi-dimensional domain, multi-dimensional functions with bounded residual term, and multi-dimensional functions that are monotone in each argument. In Section 3.5 we show some examples of our results. In Section 3.6 we discuss some of the work that Homem-de Mello (2006) has done since the original paper (Drew and Homem-de-Mello, 2005) was written. This work extends our large deviations results to stochastic optimization, and in Section 3.7 we present concluding remarks.

3.2 Background

3.2.1 Large Deviations

We begin with a brief overview of some of the basic results from large deviations theory. For more comprehensive discussions, we refer to books such as Dembo and Zeitouni (1998) or den Hollander (2000).

Suppose Y is a real-valued variable with mean $\mu = \mathbb{E}[Y]$ (possibly infinite) and let $\frac{1}{n}S_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$ be an unbiased estimator of $\mathbb{E}[Y]$, where $Y_1, \ldots, Y_n$ are not necessarily i.i.d.

of Y. Define the extended real-valued function

$$\phi_n(\theta) \;:=\; \frac{1}{n}\log \mathbb{E}[\exp(\theta S_n)]. \tag{3.7}$$

It is easy to check that $\phi_n(\cdot)$ is convex with $\phi_n(0) = 0$.

Let (a, b) be an interval on the real line containing μ. We wish to calculate the probability that the estimator S_n/n deviates from μ, i.e.

$$\mathbb{P}\left(\frac{1}{n}S_n \notin (a,b)\right) = \mathbb{P}\left(\frac{1}{n}S_n \leq a\right) + \mathbb{P}\left(\frac{1}{n}S_n \geq b\right).$$

For all $\theta > 0$, it holds that $\mathbb{P}(\frac{1}{n}S_n \geq b) = \mathbb{P}(S_n \geq bn) = \mathbb{P}(\exp(\theta S_n) \geq \exp(\theta bn))$. By applying Chebyshev's inequality to the latter term we obtain $\mathbb{P}(\frac{1}{n}S_n \geq b) \leq \exp(-\theta bn)\mathbb{E}[\exp(\theta S_n)]$ and thus

$$\frac{1}{n}\log\left[\mathbb{P}\left(\frac{1}{n}S_n \geq b\right)\right] \;\leq\; -\left(\theta b - \frac{1}{n}\log \mathbb{E}[\exp(\theta S_n)]\right) \;=\; -[\theta b - \phi_n(\theta)].$$

Note that this inequality holds regardless of any independence assumptions on the Y_is. Moreover, since the above inequality is true for all $\theta \geq 0$ it follows that

$$\frac{1}{n}\log\left[\mathbb{P}\left(\frac{1}{n}S_n \geq b\right)\right] \;\leq\; \inf_{\theta\geq 0} -[\theta b - \phi_n(\theta)] \;=\; -\sup_{\theta\geq 0}[\theta b - \phi_n(\theta)]. \tag{3.8}$$

By Jensen's inequality, we have $\mathbb{E}[\exp(\theta S_n)] \geq \exp(\theta \mathbb{E}[S_n]) = \exp(\theta n\mu)$ for any $\theta \in \mathbb{R}$ and hence

$$\phi_n(\theta) \;\geq\; \theta\mu \qquad \text{for all } \theta \in \mathbb{R}. \tag{3.9}$$

It follows that $\theta b - \phi_n(\theta) \leq \theta(b - \mu)$. Since $b > \mu$, we can take the supremum in (3.8) over $\theta \in \mathbb{R}$.

Similarly, for all $\theta < 0$ it holds that $\mathbb{P}(\frac{1}{n}S_n \leq a) = \mathbb{P}(S_n \leq an) = \mathbb{P}(\exp(\theta S_n) \geq \exp(\theta an))$. By repeating the argument in the paragraphs above we conclude that

$$\frac{1}{n}\log\left[\mathbb{P}\left(\frac{1}{n}S_n \geq b\right)\right] \leq -I(n,b) \tag{3.10a}$$

$$\frac{1}{n}\log\left[\mathbb{P}\left(\frac{1}{n}S_n \leq a\right)\right] \leq -I(n,a), \tag{3.10b}$$

where the function $I(n,z)$ is defined as

$$I(n,z) \;:=\; \sup_{\theta\in\mathbb{R}}[\theta z - \phi_n(\theta)]. \tag{3.11}$$

Note that (3.10) holds for all $n \geq 1$. Also, $I(n,z) \geq 0$ for all n and all z. We would like, however, to establish that $I(n,z) > 0$ for $z \neq \mu$ and all n, in which case the deviation probabilities in (3.10) yield an exponential decay (note that, since $\theta z - \phi_n(\theta) \leq \theta(z-\mu)$ for all θ and z, it follows that $I(n,\mu) = 0$ for all n — a natural conclusion since we cannot expect to have an exponential decay for the probability $\mathbb{P}(S_n/n \geq \mu)$).

We proceed now in that direction. Suppose that the functions $\{\phi_n(\cdot)\}$ are bounded above by a common function $\phi^*(\cdot)$. Then, by defining $I^*(z) := \sup_{\theta\in\mathbb{R}}[\theta z - \phi^*(\theta)]$ we have that (3.10) holds with $I^*(a)$ and $I^*(b)$ in place of respectively $I(n,a)$ and $I(n,b)$. Since those quantities do not depend on n, it follows that $\mathbb{P}(S_n/n \leq a)$ and $\mathbb{P}(S_n/n \geq b)$ converge to zero *at least as fast as* the exponential functions $\exp(-nI^*(a))$ and $\exp(-nI^*(b))$, i.e.,

$$\mathbb{P}\left(\frac{1}{n}S_n \geq b\right) \;\leq\; \exp(-nI^*(b)), \qquad \mathbb{P}\left(\frac{1}{n}S_n \leq a\right) \;\leq\; \exp(-nI^*(a)). \tag{3.12}$$

The proposition below establishes further conditions on ϕ^* in order for I^* to have some desired properties. In particular, under those conditions I^* is a *rate function* (in the sense of Dembo and Zeitouni 1998) that satisfies $I^*(z) > 0$ for all $z \neq \mu$.

Proposition 3.1. *Consider the functions* $\{\phi_n(\cdot)\}$ *defined in* (3.7). *Suppose that there exists an extended real-valued function* $\phi^*(\cdot)$ *such that* $\phi_n(\cdot) \leq \phi^*(\cdot)$ *for all* n, *with* ϕ^* *satisfying the following properties:* (i) $\phi^*(0) = 0$; (ii) $\phi^*(\cdot)$ *is continuously differentiable and strictly convex on a neighborhood of zero; and* (iii) $(\phi^*)'(0) = \mu$.

Then, the function $I^*(z) := \sup_{\theta \in \mathbb{R}} [\theta z - \phi^*(\theta)]$ *is lower semi-continuous, convex everywhere and strictly convex on a neighborhood of* μ. *Moreover,* $I^*(\cdot)$ *is non-negative and* $I^*(\mu) = 0$.

Proof. From (3.9), we have $\phi_n(\theta) \geq \theta\mu$ for all $\theta \in \mathbb{R}$ and hence $\phi^*(\theta) \geq \theta\mu$ for all $\theta \in \mathbb{R}$. It follows from Theorem X.1.1.2 in Hiriart-Urruty and Lemarechal (1993) that I^* (the conjugate function of ϕ^*) is convex and lower semi-continuous.

Next, condition (ii) implies that $(\phi^*)'(\cdot)$ is continuous and strictly increasing on a neighborhood of zero. Since $(\phi^*)'(0) = \mu$ by condition (iii), there exists some $\varepsilon > 0$ such that, given any $z_0 \in [\mu - \varepsilon, \mu + \varepsilon]$, there exists θ_0 satisfying $(\phi^*)'(\theta_0) = z_0$. It follows from Theorem X.4.1.3 in Hiriart-Urruty and Lemarechal (1993) that I^* is strictly convex on $[\mu - \varepsilon, \mu + \varepsilon]$.

Non-negativity of $I^*(\cdot)$ follow immediately from $\phi^*(0) = 0$. Finally, since $\theta\mu - \phi^*(\theta) \leq 0$ for all $\theta \in \mathbb{R}$, it follows from the definition of I^* that $I^*(\mu) = 0$. □

A simple setting where the conditions of Proposition 3.1 are satisfied is when the functions ϕ_n are bounded by the log-moment generating function of some random variable W (i.e., $\phi^*(\theta) = \log \mathbb{E}[\exp(\theta W)]$) such that $\mathbb{E}[W] = \mu$. Clearly, condition (i) holds in that case. Moreover, if there exists a neighborhood $\mathcal{N}$ of zero such that $\phi^*(\cdot)$ is finite on $\mathcal{N}$, then it is well known that ϕ^* is infinitely differentiable on $\mathcal{N}$ and (iii) holds. In that case, Proposition 1 in Shapiro et al. (2002) ensures that ϕ^* is strictly convex on $\mathcal{N}$.

The developments above are valid regardless of any i.i.d. assumption on the samples

$\{Y_i\}$. When such an assumption is imposed, we have

$$\phi_n(\theta) \;=\; \frac{1}{n}\log(\mathbb{E}[\exp(\theta S_n)]) \;= \frac{1}{n}\log(\{\mathbb{E}[\exp(\theta Y_1)]\}^n) \;= \log(\mathbb{E}[\exp(\theta Y_1)]) \;= \log M_{Y_1}(\theta), \tag{3.13}$$

where $M_{Y_1}(\theta)$ is the moment generating function of Y_1 evaluated at θ. In that case, of course, we have $\phi_n(\theta) = \phi^*(\theta)$ for all n, and the resulting function I^* is the rate function associated with Y_1. The inequalities in (3.12) then yield the well-known Chernoff upper bounds on the deviation probabilities.

Inequalities such as (3.12), while useful in their own, do not fully characterize the deviation probabilities since they only provide an upper bound on the decay. One of the main contributions of large deviations theory is the verification that, in many cases, the decay rate given by those inequalities is *asymptotically exact*, in the sense that (3.10) holds with equality as n goes to infinity. One such case is when $\{Y_i\}$ is i.i.d.; that, of course, is the conclusion of the well-known Cramér's Theorem.

In general, the idea of an asymptotically exact decay rate is formalized as follows. The estimator $\frac{1}{n}S_n$ — calculated from possibly non-i.i.d. random variables — is said to satisfy a *large deviation principle (LDP)* with rate function $I(\cdot)$ if the following conditions hold:

1. $I(\cdot)$ is lower semi-continuous, i.e., it has closed level sets;

2. For every closed subset $F \in \mathbb{R}$,

$$\limsup_{n\to\infty} \frac{1}{n}\log \mathbb{P}\left(\frac{1}{n}S_n \in F\right) \leq -\inf_{x\in F} I(x)$$

3. For every open subset $G \in \mathbb{R}$,

$$\liminf_{n\to\infty} \frac{1}{n}\log \mathbb{P}\left(\frac{1}{n}S_n \in G\right) \geq -\inf_{x\in G} I(x).$$

$I(\cdot)$ is said to be a *good rate function* if it has compact level sets. Note that this implies that there exists some point x such that $I(x) = 0$.

As mentioned above, in the i.i.d. case an LDP holds with I^* — the rate function of Y_1 — in place of I. The main tool for the general case is the Gartner-Ellis Theorem, which we describe next. Let ϕ_n be as defined in (3.7), and define

$$\phi(\theta) := \lim_{n\to\infty} \phi_n(\theta) \quad \text{when the limit exists.} \tag{3.14}$$

Roughly speaking, the theorem asserts that, under proper conditions, a large deviation principle holds for the estimator $\frac{1}{n}S_n$, with the rate defined in terms of the limiting $\phi(\theta)$ defined in (3.14).

The assumptions of the theorem are the following:

Assumption 3.2. *For each $\theta \in \mathbb{R}$, the function $\phi(\theta)$ defined in* (3.14) *exists as an extended real number.*

Assumption 3.3. *0 belongs to the interior of D_ϕ where $D_\phi = \{\theta \in \mathbb{R} : \phi(\theta) < \infty\}$.*

Assumption 3.4. *$\phi(\theta)$ is essentially smooth, i.e. the following three conditions hold:*

i) The interior of D_ϕ is nonempty

ii) $\phi(\theta)$ is differentiable on the interior of D_ϕ

iii) Either $D_\phi = \mathbb{R}$ or $\phi(\theta)$ is steep, i.e. for $\theta \in D_\phi$ as θ approaches the boundary of D_ϕ, $|\phi'(\theta)| = \infty$ (where $\phi'(\theta)$ is the derivative of ϕ with respect to θ).

Assumption 3.5. *$\phi(\theta)$ is lower semi-continuous.*

Theorem 3.6. (Gartner-Ellis Theorem) *Suppose that Assumption 3.2 holds, and define the function $I(x) := \sup_{\theta\in\mathbb{R}} [\theta x - \phi(\theta)]$.*

1. *If Assumption 3.3 also holds, then for every closed subset F of $\mathbb{R}$,*

$$\limsup_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n} S_n \in F\right) \leq -\inf_{x\in F} I(x).$$

2. *If Assumption 3.4 holds (in addition to Assumptions 3.2-3.3), then for every open subset G of $\mathbb{R}$,*

$$\liminf_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n} S_n \in G\right) \geq -\inf_{x\in G} I(x)$$

3. *If Assumption 3.5 holds (in addition to Assumptions 3.2-3.4), then a large deviation principle holds with the good rate function $I(\cdot)$.*

Proof. See Dembo and Zeitouni (1998) or den Hollander (2000). □

Our main goal is to derive conditions under which the results above can be applied under Latin Hypercube sampling. In sections 3.3 and 3.4 we will show that, under those conditions, the upper bound (3.12) holds with the same function I^* as the standard Monte Carlo (which suggests that LHS can do no worse than i.i.d. sampling). In some cases, we will be able to apply the Gartner-Ellis Theorem to show that a large deviation principle holds. Before stating those results, we review in detail the basic ideas of LHS.

3.2.2 Latin Hypercube Sampling

Let $X = [X^1, X^2, \ldots, X^d]$ be the vector of the d input variables of a simulation and let $Y = g(X) = g(X^1, X^2, \ldots, X^d)$ be the output of the simulation. Assume that all of the dimensions are independent. Let $F_j(\cdot)$ be the marginal cumulative distribution function for X^j. Suppose the quantity of interest is $\mathbb{E}[Y]$.

One possible sampling method to estimate $\mathbb{E}[Y]$ is to randomly sample n points in the sample space (Monte Carlo sampling). For each replication i from 1 to n, Uniform(0,1) ran-

dom numbers $U_i = [U_i^1, \ldots, U_i^d]$ are generated (one per dimension) which, assuming we can use the inverse transform method, yield the input random vector $X_i = [F_1^{-1}(U_i^1), \ldots, F_d^{-1}(U_i^d)]$ and the output $Y_i = g(X_i)$.

One problem with Monte Carlo sampling is that there is no guarantee that all sections of the sample space will be equally represented; input points could be clustered in one particular region. This is, of course, a well-known issue, and a considerable body of literature — notably on *quasi-Monte Carlo* methods — exists dealing with that topic. Latin Hypercube sampling, first proposed by McKay et al. (1979), falls into that category. The method splits each dimension of the sample space into n sections (or strata) each with probability $\frac{1}{n}$, and samples one observation from each stratum. The algorithm is comprised of three steps — it generates some uniform random numbers, then some random permutations and finally these elements are put together to yield the samples. The detailed algorithm is the following:

1. *Generate uniform random numbers:*

   (a) Generate a $n \times d$ matrix U of Uniform(0,1) random numbers. Let U_i^j be the $(i,j)^{th}$ entry of this matrix.

   (b) Create another $n \times d$ matrix $V(U)$ with $(i,j)^{th}$ entry $V_i^j(U) = \frac{i-1+U_i^j}{n}$. Thus each $V_i^j(U)$ is uniform on the interval $[\frac{i-1}{n}, \frac{i}{n}]$.

2. *Generate random permutations:*

   (a) Let $P(n)$ be the set of column vectors of permutations of the numbers $(1, 2, \ldots, n)$. There are $n!$ possible permutations, each equally likely. Let $\mathcal{P}$ be the set of $n \times d$ matrices where each column (representing an input variable) is a random permutation in $P(n)$ with all columns mutually independent. There are $(n!)^d$ elements in $\mathcal{P}$, each equally likely. Index these with $k = 1, \ldots, (n!)^d$ and let K be a random index.

(b) Randomly select $\Pi(K) \in \mathcal{P}$ (i.e. the K^{th} element of $\mathcal{P}$). Let $\pi_i^j(K)$ be the $(i,j)^{th}$ entry of this matrix. Note that the permutation matrix $\Pi(K)$ is independent of the random number matrix $V(U)$.

(c) In Latin Hypercube sampling, only n of the n^d strata are sampled. The rows of the $\Pi(K)$ matrix determine which hypercubes get sampled. Let $\pi_i(K) = [\pi_i^1(K), \ldots, \pi_i^d(K)]$ be the i^{th} row of $\Pi(K)$. This corresponds to the hypercube that covers the $\pi_i^1(K)^{th}$ stratum of X^1, the $\pi_i^2(K)^{th}$ stratum of X^2, ..., and the $\pi_i^d(K)^{th}$ stratum of X^d.

3. *Determine the randomly sampled point within each hypercube.*

(a) Create matrix $Z(\omega) = Z(V,K)$ with $(i,j)^{th}$ entry $Z_i^j(\omega) = V^j_{\pi_i^j(K)}(U)$. In other words, the $(i,j)^{th}$ entry of $Z(\omega)$ corresponds to the $(\pi_i^j(K), j)^{th}$ entry of $V(U)$ based on the permutation matrix. Thus the j^{th} column $V^j(U)$ of the random number matrix $V(U)$ is permuted according to the j^{th} column of the permutation matrix $\Pi(K)$.

(b) Let $X_i^j(\omega) = F_j^{-1}[Z_i^j(\omega)]$. Then $X_i(\omega) = [X_i^1(\omega), \ldots, X_i^d(\omega)]$ and $Y_i(\omega) = g(X_i(\omega))$.

The algorithm above generates n random vectors $Z_i(\omega) = [Z_i^1(\omega), \ldots, Z_i^d(\omega)]$, each of which is uniformly distributed on $[0,1]^d$. Unlike standard Monte Carlo, of course, the vectors $Z_1, \ldots, Z_n$ are *not* independent. These vectors are mapped via inverse transform into vectors $X_1, \ldots, X_n$, which then are used to generate the samples $Y_1, \ldots, Y_n$. It is well known that each Y_i generated by the LHS method is an unbiased estimate of $\mathbb{E}[Y]$ (see, e.g., the appendix in McKay et al. 1979).

More formally, let $f : [0,1]^d \mapsto \mathbb{R}^d$ be the function that converts the uniform random vector $Z_i(\omega)$ into the random vector $X_i(\omega)$, and let $h := g \circ f$. Then we have $Y_i(\omega) =$

$g(X_i(\omega)) = g(f(Z_i(\omega))) = h(Z_i(\omega))$. Thus, without loss of generality we will assume that the outputs Y_i are functions of random vectors that are uniformly distributed on $[0,1]^d$.

3.2.3 Calculus Results

For the remaining sections of this paper, we will need to define some notation and recall some results from analysis. These results are known but we state them for later reference. The discussion below follows mostly Bartle (1987) and Royden (1988).

Let $P := \{z_0, z_1, \ldots, z_n\}$ be a partition of the interval $[a,b]$ with $a = z_0 < z_1 < \cdots < z_{n-1} < z_n = b$ and let $|P|$ denote the norm of that partition (the maximum distance between any two consecutive points of the partition). Let $h : [a,b] \mapsto \mathbb{R}$ be a *bounded* function. A *Riemann sum* of h corresponding to P is a real number of the form $R(P,h) = \sum_{i=1}^{n} h(\xi_i)(z_i - z_{i-1})$, where $\xi_i \in [z_{i-1}, z_i]$, $i = 1, \ldots, n$. Particular cases of Riemann sums are the lower and upper Riemann sums, defined by setting ξ_i respectively to $\operatorname{argmin}_{z \in [z_{i-1}, z_i]} h(z)$ and $\operatorname{argmax}_{z \in [z_{i-1}, z_i]} h(z)$. We will denote the lower and upper Riemann sums respectively by $L(P,h)$ and $U(P,h)$.

The following is an alternative definition of Riemann integrability. It is called Cauchy criterion for integrability (see, e.g., Bartle 1987, Theorem 29.4).

Definition 3.7. *Let $h : [a,b] \mapsto \mathbb{R}$ be a bounded function. The function h is* Riemann integrable *if, given $\varepsilon > 0$, there exists $\eta > 0$ such that, for any two partitions P, Q with $|P| < \eta$, $|Q| < \eta$ and any corresponding Riemann sums $R(P,h)$ and $R(Q,h)$, we have that $|R(P,h) - R(Q,h)| < \varepsilon$.*

Clearly, if h is Riemann integrable then the true integral $\int_a^b h(z)dz$ lies between the upper and lower Riemann sums, as does any other Riemann sum $R(P,h)$. Moreover, Riemann integrability ensures that $U(P,h) - L(P,h) < \varepsilon$ if $|P| < \eta$, which in turn implies that $|R(P,h) - \int_a^b h(z)dz| < \varepsilon$ for any other Riemann sum $R(P,h)$ such that $|P| < \eta$. At this

point it is worthwhile recalling that a bounded function h is Riemann integrable if and only if the set of points at which h is discontinuous has Lebesgue measure zero (Royden, 1988, p. 85).

In our setting we will often deal with functions that are *not* bounded. In that case, we say that $h : [a, b] \mapsto \mathbb{R}$ is *integrable* if h is Lebesgue integrable. Of course, when h is bounded and Riemann integrable both the Lebesgue and the Riemann integrals coincide.

The next lemma shows that for every integrable function $h : [a, b] \mapsto \mathbb{R}$ and any point $s \in (a, b)$, we can find a small interval around s such that the integral of h on that interval is arbitrarily small.

Lemma 3.8. *Suppose $h : [a, b] \mapsto \mathbb{R}$ is integrable. Then for every $\varepsilon > 0$ there exists $\delta > 0$ such that $\left|\int_{s-\delta}^{s+\delta} h(z)dz\right| < \varepsilon$ for all $s \in (a, b)$. Also, $\left|\int_{a}^{a+\delta} h(z)dz\right| < \varepsilon$ and $\left|\int_{b-\delta}^{b} h(z)dz\right| < \varepsilon$.*

Proof. Let h^+ and h^- denote respectively the positive and negative parts of h (so $h^+ \geq 0$, $h^- \geq 0$, and $h = h^+ - h^-$). Since h is integrable, so are h^+ and h^-. Fix now $\varepsilon > 0$. Then, by Proposition 4.14 in Royden (1988) there exists $\delta^+ > 0$ such that $\int_A h^+(z)dz < \varepsilon/2$ for all sets $A \subset [a, b]$ whose Lebesgue measure is less than δ^+. Similarly, there exists $\delta^- > 0$ such that $\int_A h^-(z)dz < \varepsilon/2$ for all sets $A \subset [a, b]$ whose Lebesgue measure is less than δ^-. Take now δ with $0 < \delta < \min\{\delta^+, \delta^-\}/2$. Then, for any $s \in [a, b]$ we have that

$$\left|\int_{s-\delta}^{s+\delta} h(z)dz\right| \;\leq\; \left|\int_{s-\delta}^{s+\delta} h^+(z)dz\right| + \left|\int_{s-\delta}^{s+\delta} h^-(z)dz\right| \;<\; \varepsilon/2 + \varepsilon/2 \;=\; \varepsilon.$$

Similarly, we obtain $\left|\int_{a}^{a+\delta} h(z)dz\right| < \varepsilon$ and $\left|\int_{b-\delta}^{b} h(z)dz\right| < \varepsilon$. □

3.3 The One-Dimensional Case

We study now large deviations properties of the estimators generated by LHS. In order to facilitate the analysis, we start by considering the one-dimensional case.

Let $h : [0,1] \mapsto \mathbb{R}$ be a real-valued function in one variable, and suppose we want to estimate $\mathbb{E}[h(Z)]$, where Z has Uniform(0,1) distribution. In standard Monte Carlo sampling, the samples $Z_i(\omega)$ are all independent Uniform(0,1) random variables. In that case, we have from (3.13) that

$$\phi^{MC}(\theta) \;:=\; \phi_n^{MC}(\theta) \;=\; \log(\mathbb{E}[\exp(\theta h(Z_1))]) \;=\; \log\left[\int_0^1 \exp(\theta h(z))dz\right],$$

which is independent of n.

In LHS, when the interval $[0,1]$ is split into n strata of equal probability $\frac{1}{n}$, the intervals are all of the form $[\frac{j-1}{n}, \frac{j}{n}]$ and each random variable $Z_i(\omega)$ is now uniform on some interval of length $\frac{1}{n}$. Further, independence no longer holds.

We make the following assumptions about the function $h(z) : [0,1] \mapsto \mathbb{R}$:

Assumption 3.9.

(a) *$h(\cdot)$ is an integrable function (i.e., $|\int_0^1 h(z)dz| < \infty$).*

(b) *$h(\cdot)$ has at most a finite number of singularities.*

(c) *$h(\cdot)$ has a finite moment generating function (i.e. $\int_0^1 \exp(\theta h(z))dz < \infty$ for all $\theta \in \mathbb{R}$).*

(d) *The set of points at which $h(\cdot)$ is discontinuous has Lebesgue measure zero.*

A simple situation where the assumptions above are satisfied is when $h(\cdot)$ is a bounded function with at most countably many discontinuities; however, we do allow $h(\cdot)$ to be

unbounded. Also, it can be shown that the third part of this assumption is equivalent to assuming that $D_\phi = \mathbb{R}$.

To show that LHS satisfies a large deviation principle, we will show that it satisfies the assumptions of the Gartner-Ellis Theorem. In what follows, $Z_1, \ldots, Z_n$ are samples of a Uniform(0,1) random variable, generated by the Latin Hypercube sampling algorithm, and ϕ_n^{LHS} is defined as in (3.7), with $Y_i = h(Z_i)$.

Our main result in this section is Theorem 3.13 below. Before stating that result, we introduce some auxiliary results.

Lemma 3.10. *Suppose Assumption 3.9 holds. Then,*

$$\phi_n^{LHS}(\theta) \;=\; \theta \frac{1}{n} \sum_{i=1}^{n} c_i(n), \tag{3.15}$$

where $c_i(n)$ is defined as

$$c_i(n) \;:=\; \frac{1}{\theta} \log \left(n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z)) dz \right). \tag{3.16}$$

Proof. Following the notation defined in the LHS algorithm described above, let us denote the LH samples by $Z_1(V,K), \ldots, Z_n(V,K)$. Let $\mathbb{E}_V[\cdot]$ denote the expectation with respect to the random number matrix $V = V(U)$, and $\mathbb{E}[\cdot]$ with no subscripts denote the expectation with respect to both V and K. We have

$$\begin{aligned} \exp(n\phi_n^{LHS}(\theta)) \;&=\; \mathbb{E}\left[\exp(\theta S_n)\right] \;=\; \mathbb{E}\left[\exp\left(\theta \sum_{i=1}^{n} h(Z_i(V,K))\right)\right] \\ &= \sum_{k=1}^{n!} \mathbb{E}_V \left[\exp\left(\theta \sum_{i=1}^{n} h(Z_i(V,K))\right) \middle| K = k \right] \mathbb{P}(K=k). \end{aligned}$$

Since each permutation is equally likely, $\mathbb{P}(K = k) = \frac{1}{n!}$ for all permutations k. For each permutation k, one of the $Z_i(V, k)$ is uniform on stratum $[0, \frac{1}{n}]$, another is uniform on stratum $[\frac{1}{n}, \frac{2}{n}]$, etc., and every interval $[\frac{i-1}{n}, \frac{i}{n}]$ is sampled exactly once. It is easy to see then that the random variables $Z_1(V, K), \ldots, Z_n(V, K)$ are exchangeable and thus the conditional distribution of $\sum_{i=1}^{n} h(Z_i(V, K))$ on $K = k$ is the same for all k. Further, once the permutation has been fixed, the samples in each stratum are independent. It follows that

$$\begin{aligned}
\exp(n\phi_n^{LHS}(\theta)) &= \frac{1}{n!}\sum_{k=1}^{n!}\mathbb{E}_V\left[\exp\left(\theta\sum_{i=1}^{n}h(Z_i(V,K))\right)\middle|\, K=k\right] \\
&= \mathbb{E}_V\left[\prod_{i=1}^{n}\exp(\theta h(Z_i(V,K)))\middle|\, K=1\right] \\
&= \prod_{i=1}^{n}\mathbb{E}_V\left[\exp(\theta h(Z_i(V,K)))|\, K=1\right] \\
&= \prod_{i=1}^{n}\int_{\frac{i-1}{n}}^{\frac{i}{n}}\exp(\theta h(z))\, n\, dz. \qquad (3.17)
\end{aligned}$$

To get the latter equation, we have assumed (without loss of generality) that the permutation $k = 1$ is $(1, 2, \ldots, n)$. Also, we have used the fact that the density function of a Uniform$(\frac{i-1}{n}, \frac{i}{n})$ random variable Z_i is $n\, dz$.

By the finiteness of the moment generating function, $\exp(\theta h(\cdot))$ has some finite average value on the interval $[\frac{i-1}{n}, \frac{i}{n}]$, which is given by $1/(1/n)\int_{(i-1)/n}^{i/n}\exp(\theta h(z))dz$. This average value is equal to $\exp(\theta c_i(n))$, where $c_i(n)$ is defined in (3.16). Substituting the expression

above back into (3.17) we obtain

$$\exp(n\phi_n^{LHS}(\theta)) \;=\; \prod_{i=1}^{n} n\int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))dz \;=\; \prod_{i=1}^{n} \exp(\theta c_i(n)) \;=\; \exp\left(\theta \sum_{i=1}^{n} c_i(n)\right) \tag{3.18}$$

and hence

$$\phi_n^{LHS}(\theta) \;=\; \theta\frac{1}{n}\sum_{i=1}^{n} c_i(n). \qquad \square$$

Lemma 3.11. *For any $\theta \in \mathbb{R}$, the quantities $c_i(n)$ defined in Lemma 3.10 satisfy*

$$\theta\int_{\frac{i-1}{n}}^{\frac{i}{n}} h(z)dz \;\le\; \theta\frac{1}{n}c_i(n) \;\le\; \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))dz. \tag{3.19}$$

Proof. Using Jensen's inequality, we have, for each $i = 1, \ldots, n$,

$$\theta\frac{1}{n}c_i(n) \;=\; \frac{1}{n}\log\left(\int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))\, n\, dz\right) \;\ge\; \frac{1}{n}\int_{\frac{i-1}{n}}^{\frac{i}{n}} \log\left(\exp(\theta h(z))\right)\, n\, dz \;=\; \theta\int_{\frac{i-1}{n}}^{\frac{i}{n}} h(z)dz.$$

On the other hand, since $x \le \exp(x)$ for any x, we have

$$\frac{1}{n}\theta c_i(n) \;\le\; \frac{1}{n}\exp(\theta c_i(n)) \;=\; \frac{1}{n}\int_{\frac{i-1}{n}}^{\frac{i}{n}} n\exp(\theta h(z))dz \;=\; \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))dz. \qquad \square$$

The proposition below provides a key result. It shows that $\{\phi_n(\theta)\}$ converges to a *linear* function in θ.

Proposition 3.12. *Suppose Assumption 3.9 holds. Then,*

$$\lim_{n\to\infty} \phi_n^{LHS}(\theta) \;=\; \theta\int_0^1 h(z)dz. \tag{3.20}$$

Proof. Fix $\theta \in \mathbb{R}$. Our goal is to show that the limit (as $n \to \infty$) of the expression on

the right-hand side of (3.15) exists and is equal to $\theta \int_0^1 h(z)dz$. Although we do not assume that either $h(z)$ or $\exp(\theta h(z))$ is bounded, by assuming a finite number of singularities (cf. Assumption 3.9b) we can decompose the function into regions that are bounded plus neighborhoods around the singularity points.

Without loss of generality, let us assume that the function $h(z)$ has just one singularity at $z = s$ with $s \in (0, 1)$. The case with more than one singularity — but finitely many ones — is a straightforward extension of the one-singularity case. Also, if $s = 0$ the argument presented in the next paragraphs can be readily adapted by splitting the domain into two pieces, namely $[0, \delta)$ and $[\delta, 1]$ (similarly, if $s = 1$ we split the domain into $[0, 1 - \delta]$ and $(1 - \delta, 1]$).

Fix an arbitrary $\varepsilon > 0$. From Lemma 3.8, we can find a $\delta > 0$ so that both $|\int_{s-\delta}^{s+\delta} h(z)dz| \leq \varepsilon$ and $|\int_{s-\delta}^{s+\delta} \exp(\theta h(z))dz| \leq \varepsilon$. We can then split the domain into three pieces: $[0, s-\delta]$, $(s-\delta, s+\delta)$, and $[s+\delta, 1]$. Let $P(n)$ be the partition of the interval $[0, 1]$ into equal subintervals of length $\frac{1}{n}$. $|P(n)|$ is just $\frac{1}{n}$. Denote by $\tilde{P}(n) = \{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{\ell_1}{n}, s-\delta\} \bigcup \{s+\delta, \frac{\ell_2}{n}, \frac{\ell_2+1}{n}, \ldots, 1\}$ the corresponding partition of $[0, s - \delta] \bigcup [s + \delta, 1]$ with $\ell_1 = \max\{\ell \in \mathbb{N} : \frac{\ell}{n} < s - \delta\}$ and $\ell_2 = \min\{\ell \in \mathbb{N} : \frac{\ell}{n} > s + \delta\}$. Note that both ℓ_1 and ℓ_2 depend on n, but we omit the dependence to ease the notation. The partition $\tilde{P}(n)$ also has norm $\frac{1}{n}$.

The function is bounded on both $[0, s-\delta]$ and $[s+\delta, 1]$. Since $\exp(\theta h(z))$ is also bounded on these regions, it follows that for any $\left[\frac{i-1}{n}, \frac{i}{n}\right] \in [0, s - \delta]$ (or in $[s + \delta, 1]$) there exist $m_i(n) \in \left[\frac{i-1}{n}, \frac{i}{n}\right]$ and $M_i(n) \in \left[\frac{i-1}{n}, \frac{i}{n}\right]$ such that

$$\exp(\theta m_i(n)) = \min_{z \in [\frac{i-1}{n}, \frac{i}{n}]} \exp(\theta h(z))$$

$$\exp(\theta M_i(n)) = \max_{z \in [\frac{i-1}{n}, \frac{i}{n}]} \exp(\theta h(z)).$$

Then, $\exp(\theta m_i(n)) \leq \exp(\theta c_i(n)) \leq \exp(\theta M_i(n))$.

Also, without loss of generality we can assume $\theta > 0$. Then $\exp(\theta h(z))$ is a strictly increasing function in $h(z)$, and $m_i(n)$ and $M_i(n)$ are also respectively the minimum and maximum of $h(z)$ on the interval $[\frac{i-1}{n}, \frac{i}{n}]$, so $m_i(n) \leq c_i(n) \leq M_i(n)$. (If $\theta < 0$, $\exp(\theta h(z))$ is a strictly decreasing function in $h(z)$ and $m_i(n)$ and $M_i(n)$ are reversed.)

Further, define

$$\tilde{c}_{\ell_1+1}(n) \;:=\; \frac{1}{\theta}\log\left(\frac{1}{(s-\delta)-\frac{\ell_1}{n}}\int_{\frac{\ell_1}{n}}^{s-\delta}\exp(\theta h(z))dz\right)$$

and

$$\tilde{c}_{\ell_2}(n) \;:=\; \frac{1}{\theta}\log\left(\frac{1}{\frac{\ell_2}{n}-(s+\delta)}\int_{s+\delta}^{\frac{\ell_2}{n}}\exp(\theta h(z))dz\right),$$

so that $\exp(\theta\tilde{c}_{\ell_1+1}(n))$ and $\exp(\theta\tilde{c}_{\ell_2}(n))$ are equal to the average value of $\exp(\theta h(\cdot))$ on the intervals $[\frac{\ell_1}{n}, s-\delta]$ and $[s+\delta, \frac{\ell_2}{n}]$ respectively. Define now the sum

$$R(n) \;:=\; \frac{1}{n}\left[\sum_{i=1}^{\ell_1} c_i(n) + \sum_{i=\ell_2+1}^{n} c_i(n)\right] + \left((s-\delta)-\frac{\ell_1}{n}\right)\tilde{c}_{\ell_1+1}(n) + \left(\frac{\ell_2}{n}-(s+\delta)\right)\tilde{c}_{\ell_2}(n).$$

Assumption 3.9d, together with boundedness of h on $[0, s-\delta] \bigcup [s+\delta, 1]$, implies Riemann integrability of h on that region. Definition 3.7, together with the construction of $c_i(n)$ and $\tilde{c}_j(n)$, then ensures that we can find some $n > 0$ such that

$$\left|R(n) - \left(\int_0^{s-\delta} h(z)dz + \int_{s+\delta}^{1} h(z)dz\right)\right| \;\leq\; U(Q_n, h) - L(Q_n, h) \;<\; \varepsilon \tag{3.21}$$

for any partition Q_n of $[0, s-\delta] \bigcup [s+\delta, 1]$ such that $|Q_n| < 1/n$. Thus,

$$\begin{aligned}
&\left|\frac{1}{n}\sum_{i=1}^{n} c_i(n) - \int_0^1 h(z)dz\right| \\
&\quad= \left|R(n) - \left[\int_0^{s-\delta} h(z)dz + \int_{s+\delta}^1 h(z)dz\right] + \left[\frac{1}{n}c_{\ell_1+1}(n) - \left((s-\delta) - \frac{\ell_1}{n}\right)\tilde{c}_{\ell_1+1}(n)\right]\right. \\
&\qquad\left. + \left[\frac{1}{n}c_{\ell_2}(n) - \left(\frac{\ell_2}{n} - (s+\delta)\right)\tilde{c}_{\ell_2}(n)\right] + \left[\frac{1}{n}\sum_{i=\ell_1+2}^{\ell_2-1} c_i(n) - \int_{s-\delta}^{s+\delta} h(z)dz\right]\right| \\
&\quad\leq \left|R(n) - \left(\int_0^{s-\delta} h(z)dz + \int_{s+\delta}^1 h(z)dz\right)\right| + \left|\frac{1}{n}c_{\ell_1+1}(n)\right| \\
&\qquad + \left|\left((s-\delta) - \frac{\ell_1}{n}\right)\tilde{c}_{\ell_1+1}(n)\right| + \left|\frac{1}{n}c_{\ell_2}(n)\right| + \left|\left(\frac{\ell_2}{n} - (s+\delta)\right)\tilde{c}_{\ell_2}(n))\right| \\
&\qquad + \left|\frac{1}{n}\sum_{i=\ell_1+2}^{\ell_2-1} c_i(n)\right| + \left|\int_{s-\delta}^{s+\delta} h(z)dz\right|
\end{aligned}$$

The first term on the right-hand side of the inequality above is less than ε for n large enough (cf. (3.21)). Moreover, Lemma 3.11, together with Assumption 3.9 and Lemma 3.8, shows that, for each i, we have that $-\varepsilon \leq \frac{1}{n}c_i(n) \leq \varepsilon/\theta$ for n large enough, that is, $|\frac{1}{n}c_i(n)| \leq \varepsilon/\min\{\theta, 1\}$. A similar argument holds for the $\tilde{c}_j(n)$ terms. Also, Lemma 3.11 ensures that

$$\frac{1}{n}\sum_{i=\ell_1+2}^{\ell_2-1} c_i(n) \;\leq\; \frac{1}{\theta}\sum_{i=\ell_1+2}^{\ell_2-1}\int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))\,dz \;=\; \frac{1}{\theta}\int_{\frac{\ell_1+1}{n}}^{\frac{\ell_2-1}{n}} \exp(\theta h(z))\,dz \tag{3.22}$$

and

$$\frac{1}{n}\sum_{i=\ell_1+2}^{\ell_2-1} c_i(n) \;\geq\; \sum_{i=\ell_1+2}^{\ell_2-1}\int_{\frac{i-1}{n}}^{\frac{i}{n}} h(z)\,dz \;=\; \int_{\frac{\ell_1+1}{n}}^{\frac{\ell_2-1}{n}} h(z)\,dz. \tag{3.23}$$

By construction of ℓ_1 and ℓ_2, we have that $\left[\frac{\ell_1+1}{n}, \frac{\ell_2-1}{n}\right] \subseteq [s-\delta, s+\delta]$ for any n. Hence, inequalities (3.22) and (3.23), combined with Assumption 3.9 and Lemma 3.8, imply that $|\frac{1}{n}\sum_{i=\ell_1+2}^{\ell_2-1} c_i(n)| \leq \varepsilon/\min\{\theta, 1\}$ for any n.

Finally, recall that δ was chosen so that $|\int_{s-\delta}^{s+\delta} h(z)dz| \leq \varepsilon$ and $|\int_{s-\delta}^{s+\delta} \exp(\theta h(z))dz| \leq \varepsilon$.

It follows from the above developments that for n large enough we have

$$\left|\frac{1}{n}\sum_{i=1}^{n} c_i(n) - \int_0^1 h(z)dz\right| \;\leq\; 2\varepsilon + 5\varepsilon/\min\{\theta, 1\}.$$

When $\theta < 0$, the same argument laid out above can be used, replacing $\min\{\theta, 1\}$ with $\min\{-\theta, 1\}$ (when $\theta = 0$ (3.20) holds trivially). Since ε was chosen arbitrarily, it follows that $\theta\frac{1}{n}\sum_{i=1}^{n} c_i(n) \to \theta\int_0^1 h(z)dz$ as we wanted to show. □

The main result of this section is the following:

Theorem 3.13. *Let $h(z) : [0,1] \mapsto \mathbb{R}$ and suppose that Assumption 3.9 holds. Let Z be a Uniform(0,1) random variable and define $\mu_1 := \mathbb{E}[h(Z)] = \int_0^1 h(z)dz$. Then, the LHS estimator of μ_1 satisfies a large deviation principle with good rate function*

$$I^{LHS}(x) = \begin{cases} \infty, & \text{if } x \neq \mu_1 \\ 0, & \text{if } x = \mu_1. \end{cases}$$

Proof. Proposition 3.12 ensures that Assumption 3.2 holds. Let $\phi(\theta)$ denote the linear function $\theta\int_0^1 h(z)dz = \theta\mu_1$. By Assumption 3.9(a), $h(z)$ is integrable, so μ_1 is finite and $D_\phi = \mathbb{R}$. Thus the interior of D_ϕ is also $\mathbb{R}$, meaning that Assumption 3.3 holds. Also, since $\phi(\theta)$ is a linear function of θ, it is differentiable everywhere and continuous. Thus, Assumptions 3.4 and 3.5 also hold and the Gartner-Ellis Theorem (Theorem 3.6) can be fully applied. The resulting rate function is

$$I^{LHS}(x) \;=\; \sup_\theta\left[\theta x - \phi(\theta)\right] \;=\; \sup_\theta\left[\theta(x-\mu_1)\right] \;=\; \begin{cases} \infty, & \text{if } x \neq \mu_1 \\ 0, & \text{if } x = \mu_1 \end{cases}$$

which is a good rate function since $\{x : I^{LHS}(x) \leq \alpha\} = \{0\}$ for any $\alpha \geq 0$. □

Theorem 3.13 implies that, for any closed subset F of $\mathbb{R}$, as long as $\mu_1 \notin F$ we have that

$$\limsup_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n} S_n \in F\right) \leq -\inf_{x\in F} I^{LHS}(x) = -\infty.$$

That is, we have an *infinite decay rate*, as opposed to the exponential rate obtained with standard Monte Carlo. This shows that, asymptotically, LHS is much more precise than Monte Carlo.

The next result suggests that superiority of LHS (in the context of deviation probabilities) in fact holds for any finite n.

Proposition 3.14. *Consider the setting of Theorem 3.13. Let $I^{MC}(x)$ and $I^{LHS}(n,x)$ denote the (non-asymptotic) functions defined in* (3.11) *respectively for Monte Carlo and for LHS. Then, for* any *sample size n and all x we have that $I^{LHS}(n,x) \geq I^{MC}(x)$.*

Proof. For LHS we have, from Lemma 3.10,

$$\begin{aligned}
\phi_n^{LHS}(\theta) &= \theta \frac{1}{n} \sum_{i=1}^{n} c_i(n) = \frac{1}{n} \sum_{i=1}^{n} \log\left(n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z))\, dz\right) \\
&\leq \log\left[\frac{1}{n} \sum_{i=1}^{n} n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h(z)) dz\right] = \log\left[\int_0^1 \exp(\theta h(z)) dz\right] = \phi^{MC}(\theta),
\end{aligned}$$

where the inequality follows from Jensen's inequality. Thus, $\phi_n^{LHS}(\theta) \leq \phi^{MC}(\theta)$ for all n and θ. Equivalently, $I^{LHS}(n,x) \geq I^{MC}(x)$ for all x. □

Note that the result above fits the framework of the discussion around (3.12), i.e., the upper bound for the probability of a large deviation is smaller under Latin Hypercube sampling than under Monte Carlo sampling for any sample size n. Although in general a comparison of upper bounds is not particularly useful, the importance of Proposition 3.14 lies in the fact that the Monte Carlo upper bound is *tight* asymptotically. This suggests that even for small

sample sizes the deviation probabilities under LHS may be smaller than under Monte Carlo — a fact that is corroborated in the examples of Section 3.5.

3.4 The Multi-Dimensional Case

We consider now the multi-dimensional case $h : [0,1]^d \mapsto \mathbb{R}$. That is, we want to estimate $\mathbb{E}[h(Z)]$, where $Z = (Z^1, \ldots, Z^d)$ is uniformly distributed on $[0,1]^d$. We assume that the components of Z are mutually independent. As before, let $Z_1, \ldots, Z_n$ denote samples from the vector Z, so that $Z_i(\omega) = [Z_i^1(\omega), \ldots, Z_i^d(\omega)]$.

For Monte Carlo sampling, a large deviation principle holds, and we can show using a similar calculation to the one-dimensional case that the function ϕ_n defined in (3.7) is equal to

$$\phi^{MC}(\theta) = \log \left[\int_{[0,1]^d} \exp(\theta h(z)) dz \right]. \tag{3.24}$$

Again, we would like to show that a large deviation principle holds for Latin Hypercube sampling in the multi-dimensional case and that the upper bound for the probability of a large deviation under LHS is lower than it is for Monte Carlo sampling. While these assertions may not be true in general for multidimensional functions, we will focus on three special cases: (1) $h(\cdot)$ is a *separable* function, (2) $h(\cdot)$ has a bounded residual term in its ANOVA decomposition, and (3) $h(\cdot)$ is a multi-dimensional function which is monotone in each component.

In the multi-dimensional case, each Latin Hypercube permutation is equally likely with probability $\mathbb{P}(K = k) = \frac{1}{(n!)^d}$ (recall that the permutation matrices are indexed by k, and that K is a random index). As in the one-dimensional case, given a particular permutation $\Pi(k)$, the point sampled from each strata is independent of the point sampled from any

other strata, so the product and the expectation can be switched. Thus, we can write

$$\begin{aligned}\exp(n\phi_n^{LHS}(\theta)) &= \mathbb{E}\left[\prod_{i=1}^{n}\exp(\theta h(Z_i(V,K)))\right] \\ &= \sum_{k=1}^{(n!)^d}\mathbb{E}\left[\prod_{i=1}^{n}\exp(\theta h(Z_i(V,K)))\middle| K=k\right]\mathbb{P}(K=k) \\ &= \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{i=1}^{n}\mathbb{E}\left[\exp(\theta h(Z_i(V,K)))|\, K=k\right]. \end{aligned} \tag{3.25}$$

Also, given a particular permutation index k, for each sample i we have that

$$Z_i(V,k) \in \left[\frac{\pi_i^1(k)-1}{n}, \frac{\pi_i^1(k)}{n}\right] \times \cdots \times \left[\frac{\pi_i^d(k)-1}{n}, \frac{\pi_i^d(k)}{n}\right],$$

where the $\pi_i^j(k)$ indicate which strata are sampled, as defined in Section 3.2.2.

For notational convenience, define $a_i^j(k) := \frac{\pi_i^j(k)-1}{n}$ and $b_i^j(k) := \frac{\pi_i^j(k)}{n}$. Also, let $z := (z^1,\ldots,z^d)$ and $dz := dz^1\cdots dz^d$. Note that $Z_i^j(V,k)$ is uniformly distributed on the interval $(a_i^j(k), b_i^j(k))$. Then, (3.25) becomes

$$\exp(n\phi_n^{LHS}(\theta)) = \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{i=1}^{n} n^d \int_{a_i^d(k)}^{b_i^d(k)}\cdots\int_{a_i^1(k)}^{b_i^1(k)}\exp(\theta h(z))dz. \tag{3.26}$$

We now specialize the calculations for the three cases mentioned above.

3.4.1 Case 1: The Separable Function Case

We shall consider initially the case where the function h is *separable*, i.e., there exist one-dimensional functions $h^1,\ldots,h^d$ such that $h(z^1,\ldots,z^d) = h^1(z^1)+\ldots+h^d(z^d)$. Note that this is equivalent to saying that the ANOVA decomposition of h (cf. (3.2)) has residual part

equal to zero. Clearly, when h is separable we have

$$\int_{[0,1]^d} h(z)dz = \int_0^1 h^1(z^1)dz^1 + \cdots + \int_0^1 h^d(z^d)dz^d.$$

Since a separable multidimensional function can be decomposed into a sum of one dimensional functions, it is intuitive that our results from the one-dimensional case can be extended to this case. The theorem below states precisely that:

Theorem 3.15. *Suppose $h(z) : [0,1]^d \mapsto \mathbb{R}$ is a separable function and that each component h^j of h satisfies Assumption 3.9. Let Z be a random vector with independent components uniformly distributed on $[0,1]^d$, and define $\mu_d := \mathbb{E}[h(Z)] = \int_{[0,1]^d} h(z)dz$. Then, the LHS estimator of μ_d satisfies a large deviation principle with good rate function*

$$I^{LHS}(x) = \begin{cases} \infty, & \text{if } x \neq \mu_d \\ 0, & \text{if } x = \mu_d. \end{cases}$$

Proof. As in the proof of Theorem 3.13, the basic idea is to show that the functions $\{\phi_n^{LHS}(\theta)\}$ converge to the linear function $\phi^{LHS}(\theta) := \theta\mu_d$.

Using the special property of Latin Hypercube Sampling that in each dimension, each

stratum is sampled from exactly once, we get:

$$\begin{aligned}
\exp(n\phi_n^{LHS}(\theta)) &= \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{i=1}^{n} n^d \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} \exp(\theta h(z))\, dz \\
&= \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{j=1}^{d}\prod_{i=1}^{n} n \int_{a_i^j(k)}^{b_i^j(k)} \exp(\theta h^j(z^j))\, dz^j \\
&= \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{j=1}^{d}\prod_{i=1}^{n} n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h^j(z^j))\, dz^j \\
&= \prod_{j=1}^{d}\prod_{i=1}^{n} n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \exp(\theta h^j(z^j))\, dz^j \\
&= \exp\left(\theta \sum_{i=1}^{n} c_i^1(n)\right) \cdots \exp\left(\theta \sum_{i=1}^{n} c_i^d(n)\right) \\
&= \exp\left(\theta \sum_{j=1}^{d}\sum_{i=1}^{n} c_i^j(n)\right),
\end{aligned}$$

where the $c_i^j(n)$ are defined as in (3.16) (with h^j in place of h). Then,

$$\phi_n^{LHS}(\theta) = \sum_{j=1}^{d} \theta \frac{1}{n}\sum_{i=1}^{n} c_i^j(n).$$

This is just the sum of d one-dimensional cases. Thus, it follows from Proposition 3.12 that

$$\phi^{LHS}(\theta) \;=\; \lim_{n\to\infty} \phi_n^{LHS}(\theta) \;=\; \theta \sum_{j=1}^{d} \int_0^1 h^j(z^j) dz^j \;=\; \theta \int_{[0,1]^d} h(z) dz = \theta\mu_d,$$

so Assumption 3.2 holds. Since $\phi^{LHS}(\cdot)$ is a linear function, Assumptions 3.3, 3.4, and 3.5

follow and the Gartner-Ellis Theorem can be applied. We obtain

$$I^{LHS}(x) \;=\; \sup_{\theta\in\mathbb{R}}\left[\theta x-\phi^{LHS}(\theta)\right] \;=\; \sup_{\theta\in\mathbb{R}}\left[\theta(x-\mu_d)\right] \;=\; \begin{cases}\infty, & \text{if } x\neq\mu_d\\ 0, & \text{if } x=\mu_d,\end{cases}$$

which is a good rate function since there exists a point for which $I^{LHS}(x)=0$. $\square$

As before, for any closed subset F of $\mathbb{R}$, as long as $\mu_d\notin F$ we have a decay with infinite rate, i.e.,

$$\limsup_{n\to\infty}\frac{1}{n}\log\mathbb{P}\left(\frac{1}{n}S_n\in F\right)\leq-\inf_{x\in F}I^{LHS}(x)=-\infty.$$

Moreover, as in the one-dimensional case, when $h(z)$ is separable the upper bound for the probability of a large deviation is smaller under Latin Hypercube sampling than under Monte Carlo sampling for any sample size n, i.e., we have an extension of Proposition 3.14:

Proposition 3.16. *Consider the setting of Theorem 3.15. Let $I^{MC}(x)$ and $I^{LHS}(n,x)$ denote the (non-asymptotic) functions defined in* (3.11) *respectively for Monte Carlo and for LHS. Then, for* any *sample size n and all x we have that $I^{LHS}(n,x)\geq I^{MC}(x)$.*

Proof. As in the one-dimensional case, it again suffices to show that $\phi_n^{LHS}(\theta)\leq\phi^{MC}(\theta)$ for all n and θ. In the separable case, can also write $\phi_n^{LHS}(\theta)=\sum_{j=1}^d\phi_n^{(j)LHS}(\theta)$.

For Monte Carlo sampling, using a similar calculation we get,

$$\begin{aligned}\phi^{MC}(\theta) \;&=\; \log\left(\int_{[0,1]^d}\exp(\theta h(z))dz\right) \;=\; \log\left(\prod_{j=1}^d\int_0^1\exp(\theta h^j(z))dz^j\right)\\ &=\; \sum_{j=1}^d\log\left[\int_0^1\exp(\theta h^j(z))dz^j\right] \;=\; \sum_{j=1}^d\phi^{(j)MC}(\theta).\end{aligned}$$

Since we have already shown in one dimension that, for each j, $\phi_n^{(j)LHS}(\theta)\leq\phi^{(j)MC}(\theta)$ for all θ and all n, it follows that $\phi_n^{LHS}(\theta)\leq\phi^{MC}(\theta)$ for all θ and n if $h(z)$ is a separable function.

□

3.4.2 Case 2: The Bounded Residual Case

We now turn to the case where $h(\cdot)$ is not separable, but its residual term in the ANOVA decomposition is bounded. Recall that we can decompose h as $h(z) = \mu + h_{add}(z) + h_{resid}(z)$, where $\mu = \mathbb{E}[h(Z)]$, $h_{add}(z) = \sum_{j=1}^{d} h^j(z^j)$ and $\mathbb{E}[h^j(Z^j)] = \mathbb{E}[h_{resid}(Z)] = 0$ for all j. The bounded residual case assumes that $-m \leq h_{resid}(\cdot) \leq M$ where $m, M \geq 0$. This includes a large class of functions including all functions that are bounded themselves. Of course, the results below are useful only in case the bounds on the residual are significantly smaller than the bounds on the whole function, i.e. the function may not be separable but must have a strong additive component.

The infinite rate of decay for the deviation probability no longer holds in general when the separability condition is removed. However, we will show that the infinite rate does still hold if the deviation is sufficiently large. This is stated in the proposition below:

Proposition 3.17. *Suppose $h(z) : [0,1]^d \mapsto \mathbb{R}$ is a function such that its residual component satisfies $-m \leq h_{resid}(\cdot) \leq M$ for some $m, M \geq 0$. Suppose also that each term h^j of the additive component h_{add} satisfies Assumption 3.9. Let Z be a random vector with independent components uniformly distributed on $[0,1]^d$, and define $\mu := \mathbb{E}[h(Z)] = \int_{[0,1]^d} h(z)dz$.*

Then, for any a, b such that $a < \mu < b$, the LHS estimator S_n/n of μ satisfies

$$\lim_{n\to\infty} \frac{1}{n} \log \left[\mathbb{P}\left(\frac{1}{n} S_n \geq b \right) \right] = -\infty \quad \text{if } b > \mu + M \tag{3.27}$$

$$\lim_{n\to\infty} \frac{1}{n} \log \left[\mathbb{P}\left(\frac{1}{n} S_n \leq a \right) \right] = -\infty \quad \text{if } a < \mu - m. \tag{3.28}$$

Proof. By assumption, we have

$$\mu + \sum_{j=1}^{d} h^j(z^j) - m \;\leq\; h(z) \;\leq\; \mu + \sum_{j=1}^{d} h^j(z^j) + M.$$

For $\theta > 0$,

$$\exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) - m\right]\right) \;\leq\; \exp(\theta h(z)) \;\leq\; \exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) + M\right]\right).$$

Also, by the properties of the integral, for any $i \in 1, 2, \ldots, n$ and any permutation index k, we have that

$$\begin{aligned}
&\int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} n^d \exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) - m\right]\right) dz \\
&\qquad\leq \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} n^d \exp(\theta h(z))\, dz \\
&\qquad\leq \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} n^d \exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) + M\right]\right) dz.
\end{aligned}$$

Then, since all of these integrals are positive, we have

$$\begin{aligned}
&\frac{1}{(n!)^d} \sum_{k=1}^{(n!)^d} \prod_{i=1}^{n} \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} n^d \exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) - m\right]\right) dz \qquad (3.29) \\
&\qquad\leq \exp(n\phi_n(\theta)) \\
&\qquad\leq \frac{1}{(n!)^d} \sum_{k=1}^{(n!)^d} \prod_{i=1}^{n} \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} n^d \exp\left(\theta\left[\mu + \sum_{j=1}^{d} h^j(z^j) + M\right]\right) dz,
\end{aligned}$$

where we know from (3.26) that

$$\exp(n\phi_n^{LHS}(\theta)) = \frac{1}{(n!)^d}\sum_{k=1}^{(n!)^d}\prod_{i=1}^{n} n^d \int_{a_i^d(k)}^{b_i^d(k)} \cdots \int_{a_i^1(k)}^{b_i^1(k)} \exp(\theta h(z))dz.$$

By manipulating (3.29) as in the proof of Theorem 3.15 we obtain

$$\begin{aligned}
&\prod_{i=1}^{n}\exp(\theta(\mu-m))\prod_{j=1}^{d}\int_{\frac{i-1}{n}}^{\frac{i}{n}} n\exp\left(\theta h^j(z^j)\right)\, dz^j \\
&\qquad\leq\ \exp(n\phi_n(\theta)) \\
&\qquad\leq\ \prod_{i=1}^{n}\exp(\theta(\mu+M))\prod_{j=1}^{d}\int_{\frac{i-1}{n}}^{\frac{i}{n}} n\exp\left(\theta h^j(z^j)\right)\, dz^j.
\end{aligned}$$

By defining quantities $c_i^j(n)$ as in (3.16) (with h^j in place of h) we can rewrite the above inequalities as

$$\prod_{i=1}^{n}\left[\exp(\theta(\mu-m))\prod_{j=1}^{d}\exp(\theta c_i^j(n))\right] \leq \exp(n\phi_n(\theta)) \leq \prod_{i=1}^{n}\left[\exp(\theta(\mu+M))\prod_{j=1}^{d}\exp(\theta c_i^j(n))\right],$$

which further simplifies to

$$\exp\left(\theta\sum_{j=1}^{d}\sum_{i=1}^{n}c_i^j(n)+\theta n(\mu-m)\right) \leq \exp(n\phi_n(\theta)) \leq \exp\left(\theta\sum_{j=1}^{d}\sum_{i=1}^{n}c_i^j(n)+\theta n(\mu+M)\right),$$

and so

$$\frac{1}{n}\theta\sum_{j=1}^{d}\sum_{i=1}^{n}c_i^j(n)+\theta(\mu-m) \leq \phi_n(\theta) \leq \frac{1}{n}\theta\sum_{j=1}^{d}\sum_{i=1}^{n}c_i^j(n)+\theta(\mu+M). \tag{3.30}$$

Note that, as shown in the proof of Proposition 3.12, the term $\frac{1}{n}\sum_{j=1}^{d}\sum_{i=1}^{n}c_i^j(n)$ converges

to $\int_{[0,1]^d} h_{add}(z)\,dz = \mathbb{E}[h_{add}(Z)]$, which is equal to zero. Hence, given $\varepsilon > 0$ we have that

$$\theta\,(\mu - m - \varepsilon) \;\leq\; \phi_n(\theta) \;\leq\; \theta\,(\mu + M + \varepsilon)$$

for n large enough and thus

$$\sup_{\theta \geq 0} [\theta x - \theta(\mu + M + \varepsilon)] \;\leq\; \sup_{\theta \geq 0} [\theta x - \phi_n(\theta)] \;\leq\; \sup_{\theta \geq 0} [\theta x - \theta(\mu - m - \varepsilon)]$$

for any x. Clearly, when $x > \mu + M + \varepsilon$ the left-most term is infinite. Since ε was chosen arbitrarily, it follows from (3.8) that

$$\lim_n \frac{1}{n} \log \left[\mathbb{P}\left(\frac{1}{n} S_n \geq b \right) \right] = -\infty \quad \text{if } b > \mu + M.$$

For $\theta < 0$, we can use a similar argument to conclude that

$$\lim_n \frac{1}{n} \log \left[\mathbb{P}\left(\frac{1}{n} S_n \leq a \right) \right] = -\infty \quad \text{if } a < \mu - m. \qquad \square$$

Note that this result generalizes the separable case, since in that context we have $h_{resid} \equiv 0$ and so $m = M = 0$.

3.4.3 Case 3: The Monotone Case

We move now to the case of functions that possess a certain form of monotonicity, in the specific sense defined below:

Definition 3.18. *A function $h(z) : [0,1]^d \mapsto \mathbb{R}$ is said to be* monotone *if it is monotone in each argument when the other arguments are held fixed, i.e., if for all $z \in [0,1]^d$ and all*

$j = 1, \ldots, d$ *we have that either*

$$h(z^1, \ldots, z^{j-1}, z^j, z^{j+1}, \ldots, z^d) \leq h(z^1, \ldots, z^{j-1}, w^j, z^{j+1}, \ldots, z^d) \quad \textit{for all } w^j \in [0,1],\ z^j \leq w^j$$

or

$$h(z^1, \ldots, z^{j-1}, z^j, z^{j+1}, \ldots, z^d) \geq h(z^1, \ldots, z^{j-1}, w^j, z^{j+1}, \ldots, z^d) \quad \textit{for all } w^j \in [0,1],\ z^j \leq w^j.$$

Note that this includes functions that are monotonically increasing in some arguments and monotonically decreasing in others. The relevance of this case is due to the fact that monotone functions preserve a property called negative dependence, which we for completeness we define below:

Definition 3.19. *Random variables* $Y_i, i = i \ldots n$ *are called* negatively dependent *if*

$$\mathbb{P}(Y_1 \leq y_1, \ldots, Y_n \leq y_n) \leq \mathbb{P}(Y_1 \leq y_1) \cdots \mathbb{P}(Y_n \leq y_n).$$

The following lemma from Jin et al. (2003) gives an important property of negatively dependent random variables.

Lemma 3.20. *If* $Y_i, i = 1, \ldots n$ *are nonnegative and negatively dependent and if* $\mathbb{E}[Y_i] < \infty, i = 1 \ldots n$ *and* $\mathbb{E}[Y_1 \cdot \cdots \cdot Y_n] < \infty$*, then* $\mathbb{E}[Y_1 \cdot \cdots \cdot Y_n] \leq \mathbb{E}[Y_1] \cdots \mathbb{E}[Y_n]$.

In our context, we are interested in the case where $Y_i = g(Z_i^1, \ldots, Z_i^d)$, where g is nonnegative monotone and the vectors Z_i, $i = 1, \ldots, n$ are LH samples of a Uniform$([0,1]^d)$ random vector Z. Jin et al. (2003) show that (i) Latin Hypercube samples are negatively dependent, and (ii) monotone functions preserve negative dependence. Hence, if g is monotone then the random variables $Y_1, \ldots, Y_n$ are negatively dependent. In what follows we will use such a property repeatedly.

Unfortunately, in the present case it is not clear whether we can apply the Gartner-Ellis Theorem to derive large deviations results — the reason being that we do not know if negative dependence suffices to ensure convergence of the functions $\{\phi_n^{LHS}(\theta)\}$. We must note, however, that the Gartner-Ellis Theorem only provides *sufficient* (but not necessary) conditions for the validity of a large deviation principle; that is, it is possible that a large deviation principle holds in the present case even if the assumptions of the theorem are violated. A definite answer to that question is still an open problem.

Nevertheless, we can still derive results that fit the framework of the discussion around (3.12). Proposition 3.21 below provides results that are analogous to Propositions 3.14 and 3.16, i.e., it shows that the Monte Carlo rate $I^{MC}(x)$ is dominated by $I^{LHS}(n,x)$. Jin et al. (2003) show that, when the quantity to be estimated is a *quantile*, the upper bound on a deviation probability with negatively dependent sampling is less than that from Monte Carlo sampling. Here we show a similar result but in the context of estimation of the mean.

Proposition 3.21. *Suppose $h(z) : [0,1]^d \mapsto \mathbb{R}$ is a monotone function in the sense of Definition 3.18. Let Z be a random vector with independent components uniformly distributed on $[0,1]^d$, and assume that $\mathbb{E}[\exp(\theta h(Z))] < \infty$ for all $\theta \in \mathbb{R}$. Let $I^{MC}(x)$ and $I^{LHS}(n,x)$ denote the (non-asymptotic) functions defined in* (3.11) *respectively for Monte Carlo and for LHS. Then, for* any *sample size n and all x we have that $I^{LHS}(n,x) \geq I^{MC}(x)$.*

Proof. We will show that, if h is monotone, then $\phi_n^{LHS}(\theta) \leq \phi^{MC}(\theta)$ for all n and all θ. Note initially that from (3.7) we have

$$\exp(n\phi_n^{LHS}(\theta)) \;=\; \mathbb{E}\left[\prod_{i=1}^{n} \exp(\theta h(Z_i))\right]. \tag{3.31}$$

Clearly, the monotonicity of the exponential function (in the standard sense) implies that $\exp(\theta h(\cdot))$ is a monotone function in the sense of Definition 3.18 and so $\exp(\theta h(\cdot))$ preserves

negative dependence. By assumption, $\mathbb{E}[\exp(\theta h(Z)] < \infty$ for all $\theta \in \mathbb{R}$, and hence a direct application of Cauchy-Schwarz inequality shows that $\mathbb{E}\left[\prod_{i=1}^{n} \exp(\theta h(Z_i))\right] < \infty$. Thus, we can apply Lemma 3.20 to conclude that, for any θ,

$$\mathbb{E}\left[\prod_{i=1}^{n} \exp(\theta h(Z_i))\right] \;\leq\; \prod_{i=1}^{n} \mathbb{E}\left[\exp(\theta h(Z_i))\right] \;=\; \left(\mathbb{E}\left[\exp(\theta h(Z))\right]\right)^n, \tag{3.32}$$

the latter equality being a consequence of the unbiasedness of LHS estimators. Combining (3.31) and (3.32), it follows that

$$\phi_n^{LHS}(\theta) \;\leq\; \log \mathbb{E}\left[\exp(\theta h(Z))\right] \;=\; \phi^{MC}(\theta). \qquad \square$$

3.5 Examples

We now show examples comparing the probability of a large deviation under Latin Hypercube sampling and Monte Carlo sampling on five different functions. For each function, we generated both Monte Carlo and Latin Hypercube samples for various sample sizes n ($n =$ 50, 100, 500, 1000, 5000, 10000). For each sampling method and each n, we estimated the probability that the estimator S_n/n deviates by more than 0.1% of the true mean. This was calculated by doing 1000 independent replications and counting the number of occurrences in which the sample mean was not within 0.1% of the true mean, divided by 1000.

In each graph below, the x-axis represents the different sample sizes while the y-axis shows the estimated large deviations probabilities for each sample size. Estimates for both Latin Hypercube and Monte Carlo sampling are graphed as well as the upper and lower 95% confidence intervals for each estimate (represented by the dashed lines).

Example 1: $h(z) = \log(\frac{1}{\sqrt{z_1}})$. This is a one-dimensional function with a singularity at $z_1 = 0$. Its integral on $[0, 1]$ is equal to $\frac{1}{2}$. Latin Hypercube sampling considerably outperforms

Monte Carlo sampling with a large deviation probability of essentially zero by the time $n = 5000$. Meanwhile the probability of a large deviation is still roughly 0.9 for Monte Carlo sampling with $n = 10000$. This is shown in Figure 3.1.

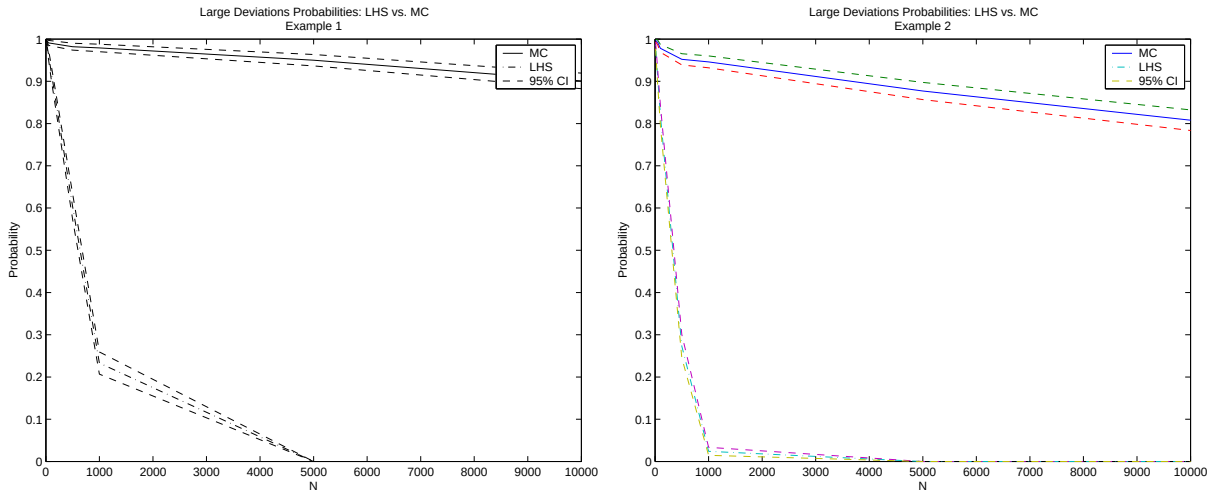

Figure 3.1: Examples 1 (left) and 2 (right).

Example 2: $h(z) = \log(z_1 z_2 z_3 z_4 z_5)$. This function is separable, so by Theorem 3.15 we expect the large deviation probability to be essentially zero under Latin Hypercube sampling with large n. The integral of the function is -5. Again the Latin Hypercube sampling dominates the Monte Carlo sampling which has a large deviation probability of nearly 0.8 at $n = 10000$. This is also shown in Figure 3.1.

Example 3: $h(z) = \log(\frac{1}{\sqrt{z_1}} + \frac{1}{\sqrt{z_2}})$. While not separable, this function is monotone in both z_1 and z_2. Its integral is $\frac{5}{4}$. From Proposition 3.21, we know that the upper bound for the large deviations probability is guaranteed to be smaller under LHS than under Monte Carlo for each value of n, and indeed we see that LHS again dominates Monte Carlo. This is shown in Figure 3.2.

Example 4: $h(z) = \log\left[2 + \sin(2\pi z_1)\cos(2\pi z_2^2)\right]$. This function is neither separable nor

monotone — in fact, it is highly non-separable. We have no guarantee that Latin Hypercube sampling will produce a lower probability of a large deviation than Monte Carlo sampling. This function has integral 0.6532 (this value was calculated numerically). From the run, we see that it is possible for Monte Carlo sampling to have a lower probability of large deviation than Latin Hypercube sampling, even at $n = 10000$. In fact, the two sampling methods give similar results for this function. This is also shown in Figure 3.2.

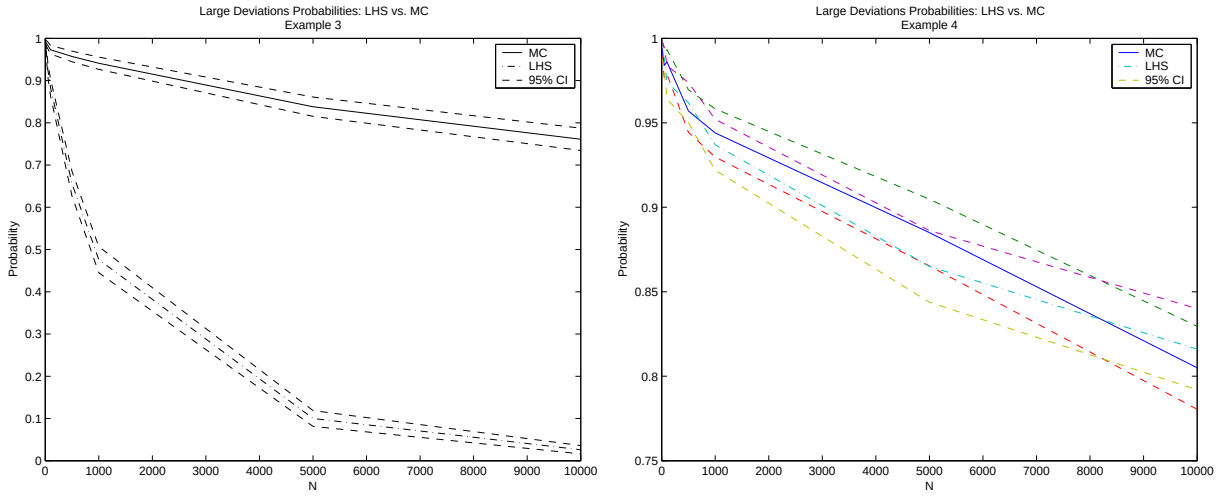

Figure 3.2: Examples 3 (left) and 4 (right).

Example 5: $h(z) = (z_1 - \frac{1}{2})^2(z_2 - \frac{1}{2})^2$. Again, this function is neither separable nor monotone. When considering deviations of 0.1% from its mean of $\frac{1}{144}$, we can see that neither deviation probability approaches zero very quickly. See Figure 3.3a. However, if we measure larger deviations such as any value of the mean outside of the interval $[0, \frac{3}{144}]$ (note that the function itself is bounded below by zero), the deviation probability approaches zero rapidly for Latin Hypercube Sampling. This is shown in Figure 3.3b for sample sizes from 1 to 10 (10000 replications). Figure 3.3c plots the logarithm of the deviation probability divided by the sample size n versus the sample size. This is the large deviation rate $-I(n, z)$. The Monte Carlo deviation rate approaches a value near -0.7 while the LHS deviation rate heads

toward $-\infty$. The graph cuts off at the point the estimated deviation probability becomes zero, in which case the logarithm cannot be calculated.

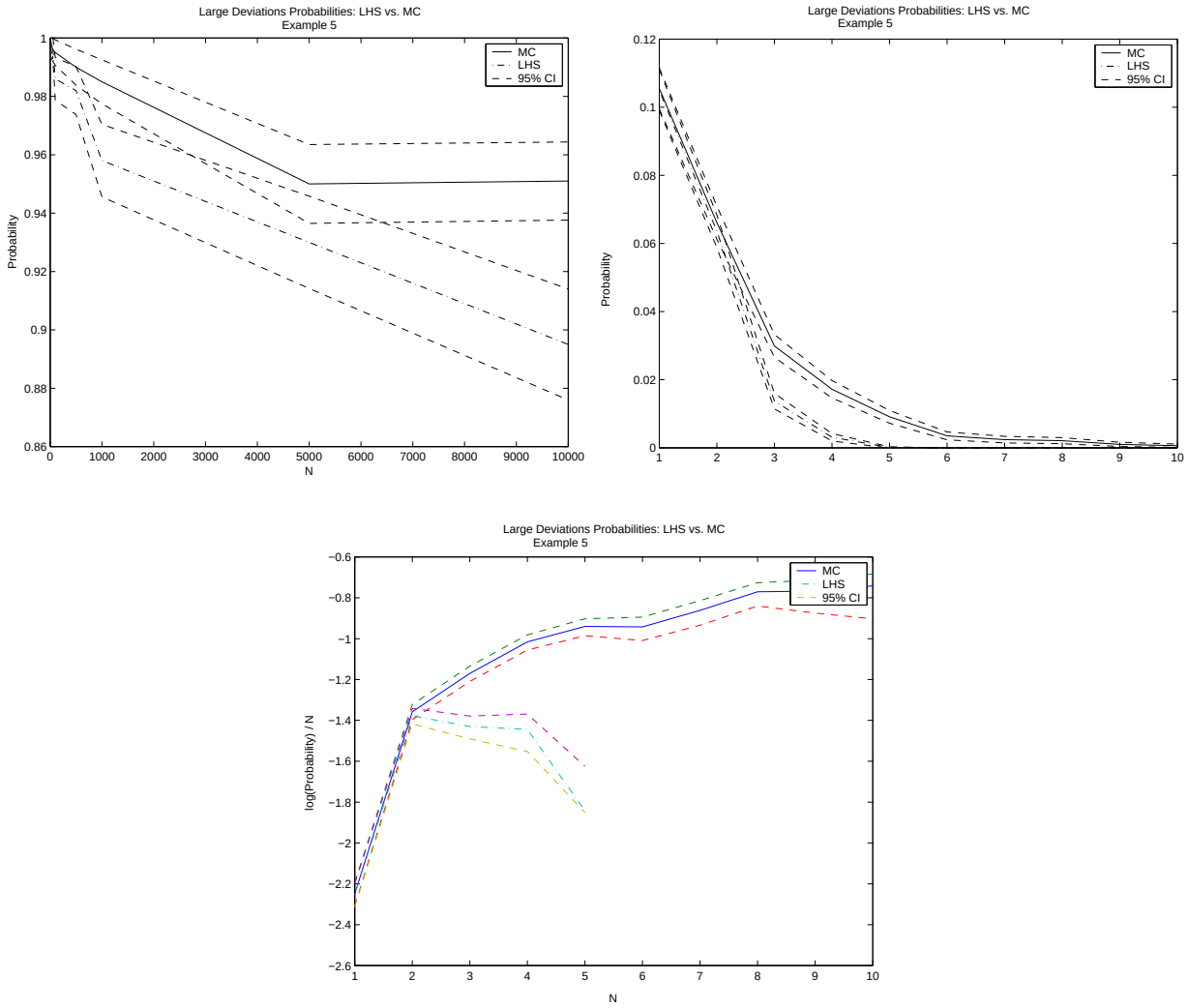

Figure 3.3: Example 5 (graphs labeled clockwise from the top left).

3.6 Connection to Stochastic Programming

Homem-de Mello (2006) has taken the large deviation results for Latin Hypercube sampling we have presented above and extended them to stochastic optimization problems.

First, we introduce a couple more assumptions:

Assumption 3.22.

1. *There exists $L > 0$ such that, for almost every ξ and all $x, y \in X$, we have*

$$|G(x,\xi) - G(y,\xi)| \leq L\|x - y\|, \tag{3.33}$$

where X is a subset of $\mathbb{R}^n$, ξ is a random vector in $\mathbb{R}^s$, and $G : \mathbb{R}^n \times \mathbb{R}^s \mapsto \mathbb{R}$ is a real valued function.

2. *The set X in Problem (1.1) is compact.* □

The first part of the assumption ensures that $G(\cdot, \xi)$ is continuous for almost every ξ and implies that the objective functions $g(x)$ and $\hat{g}_N(x)$ from (1.1) and (1.3) respectively are Lipschitz continuous with the same constant L.

The following theorem applies when the function $G(x, \xi)$ is monotone in each component of ξ.

Theorem 3.23. *Suppose that (i) Assumption 3.22 holds, (ii) $G(x, \xi)$ is monotone in each component of ξ, (iii) for each $x \in X$, the moment generating function of $G(x, \xi)$ is finite everywhere. Consider the Latin Hypercube estimators $\hat{g}_N^{LHS}(\cdot)$ and the corresponding problem $\min_{x \in X} \hat{g}_N^{LHS}(x)$. Let $\hat{x}_N^{LHS}$ denote an optimal solution of that problem and S^* be the set of optimal solutions to the true problem. Then, given $\varepsilon > 0$ there exist constants $\tilde{K} > 0$ and $\tilde{\alpha} > 0$ such that*

$$\mathbb{P}(dist(\hat{x}_N^{LHS}, S^*) \geq \varepsilon) \leq \tilde{K} e^{-\tilde{\alpha} N} \quad \text{for all } N \geq 1.$$

Moreover, the exponent $\tilde{\alpha}$ is at least as large as the corresponding exponent obtained for standard Monte Carlo.

Proof See Homem-de Mello (2006). ■

This suggests that convergence of optimal solutions under Latin Hypercube sampling will be faster that under Monte Carlo sampling. A special case of this occurs when the distribution of ξ has finite support and the true optimization problem (1.1) has a unique optimal solution x^*. Here by slightly modifying the proof of Theorem 3.2 in Shapiro and Homem-de-Mello (2000), we find that there exists $\beta > 0$ such that

$$\limsup_{N\to\infty} \frac{1}{N} \log \mathbb{P}(\hat{x}_N^{LHS} \neq x^*) = -\beta$$

and that the constant β is at least as large as the corresponding constant under Monte Carlo sampling.

When the function $G(x, \xi)$ is separable in each component of ξ, we get an even stronger result:

Theorem 3.24. *Suppose that (i) Assumption* 3.22 *holds, (ii) $G(x, \xi)$ is separable in each component of ξ, (iii) for each $x \in X$, the moment generating function of $G(x, \xi)$ is finite everywhere. Consider the Latin Hypercube estimators $\hat{g}_N^{LHS}(\cdot)$ and the corresponding problem $\min_{x\in X} \hat{g}_N^{LHS}(x)$. Let $\hat{x}_N^{LHS}$ denote an optimal solution of that problem and S^* be the set of optimal solutions to the true problem. Then, given $\varepsilon > 0$ there exist constants $\tilde{K} > 0$ and $\tilde{\alpha} > 0$ such that*

$$\mathbb{P}(dist(\hat{x}_N^{LHS}, S^*) \geq \varepsilon) \leq \tilde{K} e^{-\tilde{\alpha} N} \quad \text{for all } N \geq 1.$$

Moreover, the exponent $\tilde{\alpha}$ is at least as large as the corresponding exponent obtained for standard Monte Carlo.

In addition, we have

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{P}(dist(\hat{x}_N^{LHS}, S^*) \geq \varepsilon) = -\infty.$$

Proof Homem-de Mello (2006). ■

Thus, when the function $G(x, \xi)$ is separable in ξ, we asymptotically achieve an infinite convergence rate for the sampled optimization problem using Latin Hypercube sampling.

3.7 Conclusions

We have studied large deviations properties of estimators obtained with Latin Hypercube sampling. We have shown that LHS satisfies a large deviation principle for real-valued functions of one variable and for separable real-valued functions in multiple variables, with the rate being equal to infinity. We have also shown that the upper bound of the probability of a large deviation is smaller under LHS than it is for Monte Carlo sampling in these cases regardless of the sample size. This is analogous to the result that Latin Hypercube sampling gives a smaller variance than Monte Carlo sampling in these same cases since Var^{LHS} approaches the variance of the residual term, which in these cases is nonexistent. Further, as the number of samples becomes large, the probability of a large deviation from the true mean is essentially zero under Latin Hypercube sampling.

We have also shown that, if the underlying function is monotone in each component, then the upper bound for the large deviation probability is again less than that of Monte Carlo sampling regardless of the sample size. Again, this is analogous to the fact that the variance from LHS is no greater than that of Monte Carlo sampling when the function is monotone in all arguments. Unfortunately we do not know whether the large deviations rate is infinite, as it is in the separable case.

Large deviations results for LHS for general functions still remain to be shown, though the Latin Hypercube variance results found in the literature seem to provide a good direction. In general, the variance of a Latin Hypercube estimator may not be smaller than that of a Monte Carlo estimate (recall the bound $\mathrm{Var}^{LHS} \leq \frac{n}{n-1}\mathrm{Var}^{MC}$ proven by Owen (1997a)) ; however, asymptotically it is no worse. This might also be the case for the upper bound of the large deviations probability. Also, Stein (1987) has shown that asymptotically, Var^{LHS} is equal to just the variance of the residual term. In the separable function case, the upper bound for the large deviation probability is zero, which is also the variance of the residual term (in fact, the residual term is exactly zero). This suggests that the rate of convergence of large deviations probabilities for LHS may depend only on the residual terms — indeed, we have shown that, in case the residual term is bounded, the rate of convergence depends directly on the values of such bounds. We hope these results will stimulate further research on this topic.

Chapter 4

A Central Limit Theorem for Padded Latin Hypercube Sampling

4.1 Introduction

In this section, we consider the problem of computing $\mu = \mathbb{E}[f(X)]$ where f is a bounded function. Without loss of generality, we assume that X is uniformly distributed on the s-dimensional unit hypercube $[0,1)^s$. We can approximate μ by drawing N samples $X^1, \ldots, X^N$ from the distribution of X and replacing the expectation with the sample average $I_N = \frac{1}{N}\sum_{i=1}^{N} f(X^i)$. We then can measure the error of the sampling scheme by looking at $|I_N - \mu|$.

The most well-studied sampling scheme in this context is Monte Carlo sampling where the samples are independent and identically-distributed. It is well-known that the error from Monte-Carlo sampling is $O(\frac{1}{\sqrt{N}})$ and has an asymptotic normal distribution with a mean of 0 and some variance σ^2. Note that the order of the error is independent of the dimension of the problem. A major problem however with Monte Carlo sampling is the possibility of clustering, i.e., where the actual number of sample points in a particular region of the

sample space is much larger than the expected number. One way to remedy this problem is to deterministically choose the sample points, rather than randomly, in such a way that clustering is avoided. Such methods are called Quasi-Monte Carlo (QMC) methods. QMC methods can be further enhanced by adding a shuffling or randomization to the sampling points, which allows us to calculate errors using multiple independent replications of the sampling scheme. These enhanced methods are called Randomized QMC (RQMC) methods. A number of (R)QMC sequences have been discovered to have an error of $O(\frac{\log(N)^s}{N})$. While this is asymptotically better than the Monte Carlo rate of $O(\frac{1}{\sqrt{N}})$, the sample size N at which the QMC error becomes lower can be extremely large even for modest-sized s.

One technique that has been used to improve the performance of QMC sampling in high dimensions is called *padding*. Some subset of the random variables is sampled using a (R)QMC point set, while the rest of the variables are "padded" using some other sampling method such as the midpoint of the sampling interval, Monte Carlo sampling (Spanier 1995, Ökten 1996, Ökten 2001), or Latin Hypercube sampling (Owen, 1998).

Owen (1998) takes the concept of partitioning the random variables further with Latin Supercube sampling (LSS). Rather than partitioning the random variables into two disjoint groups (one which is sampled with a QMC point set and the other padded with another method), he partitions the random variables into r disjoint subsets, each of which gets sampled by a different point set. The run order of each point set is then permuted independently of the other point sets. It can be shown that padded sampling is a special case of LSS (Owen, 1998).

Ökten et al. (2006) show that when a QMC point set is padded with Monte Carlo sampling (a sampling scheme we will denote as PMC) to estimate the expectation of a bounded function f, then the estimator I_N has an asymptotic normal distribution, its asymptotic variance is theoretically known, and that asymptotic variance is no worse than the asymptotic

variance obtained from Monte Carlo sampling. In this chapter we show that for these same functions, when we pad with Latin Hypercube sampling (denoted PHLS), the asymptotic distribution is also normal, the asymptotic variance is also theoretically known, and that asymptotic variance is no worse than either Monte Carlo or PMC sampling.

In section 4.2, we introduce Latin Supercube sampling and show its relationship to padded sampling. We also give an expression for the variance of I_N under PHLS. In section 4.3, we show that I_N satisfies a central limit theorem under PHLS and finally in section 4.4 we provide some concluding remarks.

4.2 Latin Supercube Sampling

Latin Supercube sampling (LSS) was first introduced by Owen (1998). To construct an s-dimensional LSS point set, the random variables are partitioned into r disjoint subsets $A_1, \ldots, A_r$ with $|A_k| = s_k$ and $\sum_{k=1}^{r} s_k = s$. For the k^{th} subset or block, an s_k-dimensional (R)QMC point set $\{Q_k^i : i = 1, \ldots, N\}$ with N samples is generated. The point sets for each block are generated independently of each other. Then the run order of each point set is permuted independently of all of the other point sets. If $\{\pi_k, k = 1, \ldots, r\}$ are independent permutations of the numbers $1, \ldots, N$ for each block respectively, then the Latin Supercube sample can be written as $\{X^i : i = 1, \ldots, N\}$ with $X^i = \left[Q_1^{\pi_1(i)}, Q_2^{\pi_2(i)}, \ldots, Q_r^{\pi_r(i)}\right]$.

Latin Hypercube sampling (LHS) is just a special case of LSS where each block contains just one random variable since Latin Hypercube is indeed a QMC point set (it is a $(0, 1, s)$-net in base N). Thus, permuting the run order within each block involves simply generating a permutation for each random variable. Similarly, padding with LHS (PLHS) is also a special case of LSS. In a padded sampling scheme, $d \leq s$ random variables are sampled with a (R)QMC point set $Q = [q_1, \ldots, q_d]$ and the remaining $s-d$ random variables with a point set from some other sampling method $Y = [y_{d+1}, \ldots, y_s]$. Without loss of generality, we let the

first d components be the QMC variables, though in practice any subset of the variables may be the QMC variables. Then the padded sample is $X = [Q, Y] = [q_1, \ldots, q_d, y_{d+1}, \ldots, y_s]$. Viewing PHLS as a Latin Supercube sample yields $s - d + 1$ blocks. On the surface, PLHS does not seem to satisfy the criteria for LSS since we are not shuffling the run order within the QMC block. However, since the overall run order of the Latin Supercube sample points does not matter (we are ultimately calculating the sample mean, which is an exchangeable function, in order to estimate an integral), it suffices to shuffle all but one of the blocks.

Using the following result about the mean and variance of LSS from Owen (1998), we can determine the asymptotic variance of the PLHS estimator using the terms from the ANOVA decomposition (2.8).

For Owen's result, we must first define $\epsilon_k^N(f)$ as the worst N-sample integration error for function f obtained by the integration rule for block k when all of the components not in block k are held constant. Also, recall that an s-dimensional function f has ANOVA decomposition

$$f(\xi) = \sum_{u \subseteq S} f_u(\xi),$$

where $S = \{1, \ldots, s\}$. With this decomposition, the variance σ^2 of f can be decomposed as $\sigma^2 = \sum_{u \subseteq S} \sigma_u^2$.

Theorem 4.1. *In Latin Supercube sampling, if $\epsilon_k^N(f) = O(N^{-1/2})$, for $k = 1, \ldots, r$, then*

$$\mathbb{E}_{LSS}(I_N) = \mu + O(N^{-1/2}).$$

Also, if $\epsilon_k^N(f_u f_v) = O(N^{-1/2})$ for all $u, v \subseteq S$, and $k = 1, \ldots, r$, then

$$V_{LSS}(I_N) = \frac{1}{N}\left(\sigma^2 - \sum_{k=1}^{r} \sum_{u \subseteq A_k} \sigma_u^2\right) + O(N^{-3/2}).$$

Proof: We have modified the condition on $\epsilon_k^N(f)$ from Owen (1998) to be $O(N^{-1/2})$ rather than $o(N^{-1/2})$. This changes the orders on the remainder terms, but otherwise the proof follows directly from Owen (1998). □

It should be noted that the necessary conditions on $\epsilon_k^N(\cdot)$ hold for LHS, as well as for most QMC and RQMC sampling schemes — in particular, for (t, m, s)-nets and scrambled (t, m, s)-nets.

Theorem 4.2. *Let f be a measurable real-valued function with ANOVA decomposition $f = \sum_{u \in S} f_u$ and let $\{X^i, i = 1, \ldots, N\}$ be a PLHS point set with $X^i = [Q^i, Y^i]$ where $\{Q^i, i = 1, \ldots, N\}$ is a d-dimensional QMC point and $\{Y^i, i = 1, \ldots, N\}$ is a $(s-d)$-dimensional LHS point set. Further, let $D = \{1, \ldots, d\}$ be the set of indices of the QMC variables. Then, as the number of samples $N \to \infty$,*

1. *the asymptotic variance of I_N under PLHS is*

$$V_{PLHS}(I_N) = \frac{s_N^2}{N} = \frac{1}{N}\left(\sigma^2 - \sum_{u \subseteq D} \sigma_u^2 - \sum_{j=d+1}^{s} \sigma_{\{j\}}^2\right) + O(N^{-3/2}); \tag{4.1}$$

2. *the variance from Equation (4.1) is no larger than the variance from pure Monte Carlo sampling or from PMC sampling.*

Proof: As described above, PLHS is a special case of Latin Supercube sampling. By Theorem 1 of Owen (1998), we then have that Equation (4.1) holds.

By Theorem 7 of Ökten et al. (2006), the theoretical asymptotic variance of PMC is

$$V_{PMC}(I_N) = \frac{s_N^2}{N} = \frac{1}{N}\left(\sigma^2 - \sum_{u \subseteq D} \sigma_u^2\right) + O(N^{-3/2}) \tag{4.2}$$

and further, that variance is no larger than the asymptotic variance for pure Monte Carlo sampling, which is simply $\frac{\sigma^2}{N}$. Since, the variance terms are nonnegative, $V_{PLHS} \leq V_{PMC}$

which proves the second statement. □

We further note for reference that the asymptotic variance for pure LHS can be written as

$$V_{LHS}(I_N) = \frac{s_N^2}{N} = \frac{1}{N}\left(\sigma^2 - \sum_{j=1}^{s} \sigma_{\{j\}}^2\right) + O(N^{-3/2}). \tag{4.3}$$

4.3 A Central Limit Theorem for PLHS

To prove asymptotic normality for PLHS, we follow a similar technique to the one used in the proof of the central limit theorem for LHS (Owen, 1992). Assume that the PLHS point set in s dimensions consists of a QMC point set on indices $D = \{1, \ldots, d\}$ and a Latin Hypercube point set on the remaining dimensions $\{d+1, \ldots, s\}$. We can then define an alternate ANOVA decomposition for the function $f : [0,1)^s \to \mathbb{R}$:

$$f = \mu + \sum_{u \in D} f_u + \sum_{j=d+1} f_{\{j\}} + r \tag{4.4}$$

where r is a residual term containing all of the remaining ANOVA terms. Before, showing the central limit theorem, we first present two lemmas.

Lemma 4.3. *Let f be a bounded function over $[0,1)^s$ and r be the residual term in (4.4). Let $\mathfrak{X} \subseteq \{X^1, \ldots, X^{i-1}\}$. Then for any $i = 1, \ldots, N$:*

$$\mathbb{E}_{PLHS}[f(X^i)|\mathfrak{X}] = \mathbb{E}_{IID}[f(X^i)] + O(N^{-1}) \tag{4.5}$$

$$\mathbb{E}_{PLHS}[r(X^i)|\mathfrak{X}] = O(N^{-1}) \tag{4.6}$$

Proof: Under LHS with N samples each dimension of the sample space is split into N strata of equal probability. Within each dimension, each stratum is sampled from exactly once. Thus each sample excludes an additional area that has a volume of $O(N^{-1})$. Applying

this idea on the LHS portion of PLHS, Equation (4.5) holds because the PLHS expectation is over all of $[0,1)^i$ except for a volume of $O(N^{-1})$ excluded by the samples we know. Then since f is bounded, we can find a constant times $O(N^{-1})$ to serve for the known values in $\mathfrak{X}$.

For Equation (4.6), note that if f is bounded, its residual will also be bounded. Then Equation 4.5 holds for the residual. However, by definition of the ANOVA decomposition, the residual term has an expected value of zero, thus giving equation (4.6). $\square$

Lemma 4.4. *Let f be a bounded function over $[0,1)^s$ and let r be defined as before. Let $r_i = r(X^i)$ and $\bar{R} = \frac{1}{N}\sum_{i=1}^{N} r_i$. Then for integral $p \geq 1$*

$$\mathbb{E}_{PLHS}\{(\sqrt{N}\bar{R})^p\} = \mathbb{E}_{IID}\{(\sqrt{N}\bar{R})^p\} + O(N^{-\frac{1}{2}}) \tag{4.7}$$

as $N \to \infty$.

Proof: The term of the left side of Equation (4.7) can be rewritten as:

$$\mathbb{E}_{PLHS}\{(\sqrt{N}\bar{R})^p\} = N^{-p/2}\sum_{i_1=1}^{N}\cdots\sum_{i_p=1}^{N}\mathbb{E}_{PLHS}\left(\prod_{m=1}^{p} r_{i_m}\right).$$

We can view the terms of the resulting polynomial as follows: There are N distinguishable cells (the samples) and we need to place p indistinguishable balls into those cells. Suppose when we do this, S of the cells are nonempty. There are then $\binom{N}{S}$ ways to select the S nonempty cells, which we will index by k. Let $\{c_{jk} : j = 1,\ldots,S\}$ be the indices of the S nonempty cells and $\{a_{jk} : j = 1,\ldots,S\}$ be the number of balls in the j^{th} nonempty cell for cell choice k. Note that $\sum_{j=1}^{S} a_{jk} = p$ for all k. Then,

$$N^{-p/2}\sum_{i_1=1}^{N}\cdots\sum_{i_p=1}^{N}\mathbb{E}_{PLHS}\left(\prod_{m=1}^{p} r_{i_m}\right) = N^{-p/2}\sum_{S=1}^{p}\sum_{k=1}^{\binom{N}{S}}\sum_{a_{1k},\ldots,a_{Sk}}\mathrm{coef}_S(a_{1k},\ldots,a_{Sk})\mathbb{E}_{PLHS}\left(\prod_{j=1}^{S} r_{c_{jk}}^{a_{jk}}\right)$$

where the third sum is over all sets of positive integers $a_{1k}, \ldots, a_{Sk}$ for which $\sum_{j=1}^{S} a_{jk} = p$. Since p is fixed, the number of terms in the first and third summations does not grow with N. There are $\binom{N}{S}$ terms in the second sum which is $O(N^S)$, and so the coefficients after all of the summations will also be $O(N^S)$, whatever the values of a_{jk} as $N \to \infty$. The actual values of the coefficients are not important, just their magnitude.

It suffices to show that:

$$N^{-p/2+S}\left[\mathbb{E}_{PLHS}\left(\prod_{j=1}^{S} r_{c_{jk}}^{a_{jk}}\right) - \prod_{j=1}^{S} E_{IID}(r_{c_{jk}}^{a_{jk}})\right] = O(N^{-\frac{1}{2}}) \tag{4.8}$$

as $N \to \infty$ for all k. For readability, we will drop the k from the notation.

We handle each residual term in the product differently depending on its exponents. Residual terms with exponent 1 can be simplified using Equation (4.6) while residual terms with exponents larger than 1 must be simplified using Equation (4.5). When $a_S > 1$, by Equation (4.5) we get

$$\begin{aligned}\mathbb{E}_{PLHS}\left(\prod_{j=1}^{S} r_{c_j}^{a_j}\right) &= \mathbb{E}_{PLHS}\left[\prod_{j=1}^{S-1} r_{c_j}^{a_j}\mathbb{E}_{PLHS}(r_{c_S}^{a_S}|X_{c_1}, \ldots, X_{c_{S-1}})\right] \\ &= \mathbb{E}_{PLHS}\left(\prod_{j=1}^{S-1} r_{c_j}^{a_j}\right)\{\mathbb{E}_{IID}(r_{c_S}^{a_S}) + O(N^{-1})\},\end{aligned} \tag{4.9}$$

and when $a_S = 1$, we have by Equation (4.6) that

$$\begin{aligned}\mathbb{E}_{PLHS}\left(\prod_{j=1}^{S} r_{c_j}^{a_j}\right) &= \mathbb{E}_{PLHS}\left[\prod_{j=1}^{S-1} r_{c_j}^{a_j}\mathbb{E}_{PLHS}(r_{c_S}^{a_S}|X_{c_1}, \ldots, X_{c_{S-1}})\right] \\ &= \mathbb{E}_{PLHS}\left(\prod_{j=1}^{S-1} r_{c_j}^{a_j}\right)O(N^{-1})\},\end{aligned} \tag{4.10}$$

Suppose that none of the a_j is equal to 1. Then, they all must be at least 2, and so

$S < p/2$. Then by repeated application of Equation (4.9), the left-hand side of Equation (4.8) becomes $O(N^{-p/2+S})O(N^{-1}) = O(N^{-p/2+S-1}) = O(N^{-1}) = O(N^{-\frac{1}{2}})$.

Now, suppose that $t > 0$ of the a_j are equal to 1. Then by repeated application of Equation (4.10), the left-hand side of Equation (4.8) becomes $O(N^{-p/2+S-t})$. Now there are $S-t$ of the a_j greater than 1 and their sum is $p-t$. By a similar argument as above, $S-t \leq (p-t)/2$ and the left-hand side of Equation (4.8) is now $O(N^{-\frac{p}{2}+\frac{p-t}{2}}) = O(N^{-t/2}) = O(N^{-\frac{1}{2}})$ since $t \geq 1$. □

Theorem 4.5. *Under the conditions of Lemma 4.4, let* $Y^i = f(X^i)$ *and* $\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y^i$. *Then* $\sqrt{N}(\bar{Y} - \mu)$ *tends in distribution to* $N(0, r^2_{PLHS})$ *as* $N \to \infty$, *where*

$$r^2_{PLHS} = \left(\sigma^2 - \sum_{u \subseteq D} \sigma_u^2 - \sum_{j=d+1}^{s} \sigma^2_{\{j\}}\right).$$

Proof: The mean of $\sqrt{N}(\bar{Y} - \mu)$ is zero and the variance tends to $N \cdot V_{PLHS} = r^2_{PLHS}$ by Theorem 4.2, so it only remains to show that the limit distribution of PLHS is normal.

Using the alternate ANOVA decomposition of f in Equation (4.4), we can rewrite

$$\sqrt{N}(\bar{Y} - \mu) = \sqrt{N}\left(\sum_{u \subseteq D} \bar{f}_u + \sum_{j=d+1}^{s} \bar{f}_{\{j\}} + \bar{R}\right)$$

with $\bar{f}_u = \frac{1}{N}\sum_{i=1}^{N} f_u^i$ for both $u \subseteq D$ and $u \in \{d+1, \ldots, s\}$.

Recall that random variables $d+1, \ldots, s$ in the PLHS sample come from a Latin Hypercube point set. Thus, using a sample padded with LHS to integrate *only* over those variables is equivalent to using a pure Latin Hypercube sample. Then for $j = d+1, \ldots, s$, $\mathrm{var}_{PLHS}(\sqrt{N}\bar{f}_{\{j\}}) = \mathrm{var}_{LHS}(\sqrt{N}\bar{f}_{\{j\}}) = O(N^{-1})$ (Owen, 1992). Further, if F_j is the distribution function of random variable X_j, then $\int f_{\{j\}} dF_j = 0$ (a property of the ANOVA decomposition) and it follows that $\sqrt{N}\bar{f}_{\{j\}}$ converges in probability to 0 for $j = d+1, \ldots, s$.

Also, since the $\{f_u, u \subseteq D\}$ are the ANOVA terms corresponding only to the deterministic variables, their average value will remain the same regardless of the sample. Thus, the variance for those $\bar{f}_u$ terms is identically 0. This leaves only $\bar{R}$.

We know that a normal limit holds for $\sqrt{N}\bar{R}$ under i.i.d. sampling since the residuals are all bounded. Lemma 4.4 shows that the moments of $\sqrt{N}\bar{R}$ are asymptotically equal under i.i.d. and PLHS sampling. Then by the method of moments (Chung, 1974), a normal limit also holds for PLHS. □

4.4 Conclusions

In this chapter we have shown that padded Latin Hypercube sampling satisfies a central limit theorem for bounded functions. We have also shown that the variance under this sampling method is no worse than the variance of a comparable padded Monte Carlo sampling method. While we have given a formula for the theoretical asymptotic variance under PLHS, computing that value is often difficult in practice due to the complexity of calculating the ANOVA decomposition for functions of many variables. Additionally, there is the problem of which random variables should be chosen for the QMC block and which for the padded block. This is a question we will explore more in Chapter 5 of this dissertation.

Chapter 5

A Padded Sampling Algorithm for Stochastic Programming

5.1 Introduction

We now focus on two-stage stochastic linear programs with fixed recourse. In these problems, a decision x (from some set of decisions X) is made in the first stage in anticipation of some future random event(s). Following the random event(s), a recourse action y is then taken in the second stage. Typically these problems are of the form:

$$\min_{x \in X} \{g(x) := c^T x + \mathbb{E}[Q(x, \xi)]\} \tag{5.1}$$

where X is a subset of $\mathbb{R}^n$, the second stage stochastic program $Q(x, \xi)$ is defined as

$$Q(x, \xi) := \inf\{q^T y : Wy \geq h(\xi) - T(\xi)x,\ y \geq 0\} \tag{5.2}$$

and the random vector $\xi = (h, T)$. We also assume that all of the components of ξ are independent of each other. Thus the optimal value of the second stage stochastic program can be thought of as a function $Q(x, h, T)$. Similar to Equation (1.2) we can create a sampled optimization problem by replacing the expectation with the sample average. As before, we let $\hat{v}_N$ and $\hat{x}_N$ be the optimal objective value and some optimal solution to the sampled problem and v^* and x^* be the corresponding quantities for the true problem.

External sampling methods which solve the sampled problem using Quasi-Monte Carlo sampling have been quite successful compared to their Monte Carlo counterparts (see Section 5.2 for an overview). However, there are issues with QMC sampling. First, the upper bound for the integration error under QMC is dependent on the dimension of the problem (i.e., the number of components of ξ). Yet numerical results show that the integration error is much more in line with the effective dimension of problem instead of the larger true dimension. Second, while Quasi-Monte Carlo integration may improve on Monte Carlo integration in terms of the number of samples needed, there may not be an improvement in terms of computing time. When the dimension of the problem is large, it may become difficult and time-consuming to construct low discrepancy Quasi-Monte Carlo point sets.

One solution to this is to use a *padded* sampling scheme where only a subset of the random variables are sampled using Quasi-Monte Carlo sampling and the remaining variables are sampled using some other (typically less computationally intensive) sampling scheme such as the midpoint of the interval, a Monte Carlo sample (Spanier 1995), or a Latin Hypercube sample (Owen 1998). To maximize the efficiency of a padded sampling scheme, we can assign some measure of importance to each of the random variables, and then only choose the most important variables to be sampled with QMC. Since QMC is a method used to reduce integration error, it seems natural for each random variable's measure of importance to correspond with its contribution to the variance of the optimal value of the

optimization problem. We then can construct a smaller QMC point set on just these important components and pad the sample using some other method such as Monte Carlo or Latin Hypercube sampling on the remaining components. In Section 5.2, we review the use of sampling algorithms in stochastic optimization and in Section 5.3 we give an overview of the field of sensitivity analysis, which is the study of how the variation of the output of a model can be attributed to each of its input variables. In Section 5.4, we discuss the problem of identifying the important variables in a two-stage stochastic program with fixed recourse. We present an external sampling algorithm with padded sampling to solve two-stage stochastic programs in Section 5.5 and prove in Section 5.6 that the algorithm stops after a finite number of samples. We show some numerical results for our algorithm in Section 5.7 before offering some concluding remarks in Section 5.8.

5.2 Sampling Methods in Stochastic Optimization

5.2.1 Methods

Since the expected value in (1.1) and (5.1) is often difficult to calculate directly, sampling methods have become popular in stochastic optimization. The sampling-based algorithms tend to fall under two classes – those where the sampling procedure is executed internally to the optimization algorithm and those where the sampling procedure is executed externally.

One example of an internal method is the Stochastic Approximation (SA) algorithm introduced by Robbins and Monro (1951a). This algorithm is defined by a recursive sequence $x_{n+1} := x_n + \alpha_n \chi_n$, $n = 0, 1, \ldots$, where χ_n is the step size at iteration n, ξ_n is an estimate of the gradient $\bigtriangledown g(x_n)$ or some other random direction, and x_0 is a given starting point. The step sequence $\{\alpha_n\}$ is chosen so that it goes to zero but not too fast: $\sum_{n=0}^{\infty} \alpha_n = \infty$ and $\sum_{n=0}^{\infty} \alpha_n^2 < \infty$. Other such methods are the Stochastic Quasi-Gradient method (Ermoliev

1983) where subgradients are estimated through sampling and the Stochastic Decomposition Method (Higle and Sen 1991) where random samples are used to generate cuts at each iteration.

External sampling methods involve estimating the function $g(x)$ in (1.1) by the family of random approximations $\{\hat{g}(x_N)\}$ where $\hat{g}(x_N)$ is as defined in (1.2). These methods are sometimes referred to as Sample Average Approximations (SAA) or Sample Path Optimizations as $\hat{g}(x_N)$ is obtained by averaging the results of N sample paths $\xi^1, \ldots, \xi^N$. The external sampling approach with Monte Carlo sampling has been used in many instances, e.g., Kleywegt et al. (2001), Gürkan, Özge, and Robinson (1999), and Plambeck, Fu, Robinson, and Suri (1996). This approach has some nice convergence properties. If x^* is the unique optimal solution to the true optimization problem and v^* the optimal value and if $\hat{x}_N$ and $\hat{v}_N$ are the corresponding quantities for the sampled optimization problem, then $\hat{x}_N \to x^*$ and $\hat{v}_N \to v^*$ under some general conditions (See, e.g. Dupačová and Wets 1988, King and Rockafellar 1993, Robinson 1996, Shapiro 1991, Shapiro 1993). Shapiro (1991) has shown that the sequence of optimal objective values $\{\hat{v}_N\}$ satisfies a central limit theorem. Namely,

$$\sqrt{N}(\hat{v}_N - v^*) \Rightarrow \text{Normal}(0, \sigma_*^2)$$

where "*Rightarrow*" denotes convergence in distribution and $\sigma_*^2 := Var(G(x^*, \xi))$. Thus the convergence of optimal objective values is of order $\frac{1}{\sqrt{N}}$.

Quasi-Monte Carlo sampling (including the special case of Latin Hypercube sampling which is a $(0, 1, s)$-net in base $b = n$) has also been implemented into stochastic optimization in several places— e.g., Shapiro and Homem-de-Mello (2000) and Linderoth, Shapiro, and Wright (2005) — and numerical results indicate that the Quasi-Monte Carlo methods considerably outperform Monte Carlo methods in terms of rates of convergence of optimal values and solutions. A few papers have looked specifically at the use of QMC methods for

external sampling schemes. Kalagnanam and Diwekar (1997) show empirical results using Hammersley sequences. Koivu (2005), Pennanen and Koivu (2005), and Pennanen (2005) show that under mild assumptions when Quasi-Monte Carlo methods are used to solve (1.1), the estimator function $\hat{g}_N$ epiconverges to the true function g. This guarantees that the optimal values and optimal solutions converge with probability one. Based on the results from Chapter 3 of this thesis, Homem-de Mello (2006) has shown convergence results for deviation probabilities when Latin Hypercube sampling is used in an external sampling scheme (see section 3.6), but as of yet, there are no comparable results for general Quasi-Monte Carlo methods.

5.2.2 Stopping Criteria

The results above discuss properties of sampling methods for stochastic optimization as the number of samples N grows to ∞. Practically though, we need to determine some finite sample size at which to stop the sampling algorithm. All of the stopping criteria below require that the following assumptions hold with respect to the two-stage optimization problem in (5.1).

Assumption 5.1. *Let* $G(x,\xi) = c^T x + Q(x,\xi)$.

(A1) $G(\cdot,\xi)$*is continuous on* X

(A2) $\mathbb{E}[\sup_{x\in X} G^2(x,\xi)] < \infty$

(A3) X *is nonempty and compact.*

For instance, these assumptions will hold if (5.1) has relatively complete recourse and ξ has finite support.

Mak, Morton, and Wood (1999) introduce a stopping criteria based on the optimality gap $\mu_{\tilde{x}}$ between the true optimal value v^* and the objective value evaluated at some suboptimal

candidate solution $\tilde{x}$ obtained by the sampling algorithm. By solving multiple sampled problems, they estimate a lower bound L and an upper bound U for the true optimal value and calculate $100(1-\alpha)\%$ confidence intervals on both L and U. The lower bound comes from the result that $\mathbb{E}[\hat{v}_N] \leq \mathbb{E}[\hat{v}_{N+1}] \leq v^*$ (Mak et al., 1999). The stage 1 sampled stochastic program is solved N_L times using independent replications each with N i.i.d. samples, with replication j having optimal solution $\hat{v}_N^j$. We can then calculate:

$$L_{N,N_L} = \frac{1}{N_L} \sum_{j=1}^{N_L} \hat{v}_N^j, \tag{5.3}$$

$$s_L^2(N_L) = \frac{1}{N_L - 1} \sum_{j=1}^{N_L} (\hat{v}_N^j - L_{N,N_L})^2, \tag{5.4}$$

and

$$\epsilon_L = \frac{t_{N_L-1,\alpha} s_L(N_L)}{\sqrt{N_L}} \tag{5.5}$$

to construct the $(1-\alpha)$-confidence interval $L_{N,N_L} \pm \epsilon_L$ for the lower bound.

For an upper bound to v^*, it suffices to find a suboptimal solution to the sampled problem. Given a suboptimal stage 1 solution $\tilde{x}$, the stage 2 problem, $Q(\tilde{x}, \xi)$ (see Equation 5.2), can be solved N_U times (again with independent replications each with N i.i.d. samples). Replication j will yield suboptimal solution

$$\hat{g}_N^j(\tilde{x}) = c^T \tilde{x} + \frac{1}{N} \sum_{i=1}^{N} Q(\tilde{x}, \xi^{i,j}) = \frac{1}{N} \sum_{i=1}^{N} G(\tilde{x}, \xi^{i,j}) \tag{5.6}$$

to the sampled stochastic program. Then the upper bound for v^* can be calculated as

$$U_{N,N_U} = \frac{1}{N_U} \sum_{j=1}^{N_U} \hat{g}_N^j(\tilde{x}). \tag{5.7}$$

$s_U^2(N_U)$, ϵ_U, and the $(1-\alpha)$-confidence interval can be calculated similar to before.

They then introduce two methods of constructing confidence intervals on the optimality gap at candidate solution $\tilde{x}$. In the first method, the random number streams used to calculate L and U are independent of each other. Then

$$[0, U_{N,N_U} - L_{N,N_L} + \epsilon_L - \epsilon_U] \tag{5.8}$$

is an approximate $(1-2\alpha)$-level confidence interval for the optimality gap at $\tilde{x}$. In the second method, common random number streams are used when calculating the upper and lower bounds and the replication size is denoted $N_G = N_L = N_U$. For replication j, the gap estimate is $Gap_N^j(\tilde{x}) = \hat{g}_N^j(\tilde{x}) - \hat{v}_N^j$. The mean gap is

$$Gap_{N_G}(\tilde{x}) = \frac{1}{N_G} \sum_{j=1}^{N_G} Gap_N^j(\tilde{x}). \tag{5.9}$$

$s_{Gap}(N_G)$ and ϵ_{Gap} can be calculated similar to before and then an approximate $(1-\alpha)$-level confidence interval for the optimality gap at $\tilde{x}$ is

$$[0, Gap_{N_G}(\tilde{x}) + \epsilon_{Gap}]. \tag{5.10}$$

For both methods, the candidate solution $\tilde{x}$ is deemed optimal when the width of the confidence interval for the optimality gap at $\tilde{x}$ falls below some threshold. Bayraksan, Morton, and Partani (2007) further show that using multiple independent replications where each replication uses N non-i.i.d. samples also yields valid confidence intervals.

The methods above are called multiple replications procedures (MRP). However, in order for the MRP to have good statistical properties, the number of replications needs to be at least 30. This means we need to solve at least 30 SAA optimization problems to determine

whether or not the candidate solution is a good solution, which is not always practical.

Bayraksan and Morton (2006) develop a method to test the quality of the candidate solution $\tilde{x}$ using a single replication. We refer to the resulting single replication procedure as SRP or 1RP. As before, we start with the candidate solution $\tilde{x}$ and draw N i.i.d. samples $\xi^1, \ldots, \xi^N$ from the distribution of ξ. Next the sampled stochastic program is solved with those N samples to obtain a solution x_N^*. The gap is calculated similar to before:

$$Gap_N(\tilde{x}, x_N^*) = \frac{1}{N}\sum_{i=1}^{N}(G(\tilde{x}, \xi^i) - G(x_N^*, \xi^i)) = \bar{g}(\tilde{x}) - \bar{g}(x_N^*). \tag{5.11}$$

The sample variance is then

$$s_N^2(\tilde{x}, x_N^*) = \frac{1}{N-1}\sum_{i=1}^{N}((G(\tilde{x}, \xi^i) - G(x_N^*, \xi^i)) - (\bar{g}(\tilde{x}) - \bar{g}(x_N^*)))^2 \tag{5.12}$$

and $\epsilon_{Gap} = \frac{t_{N-1,\alpha} s_N(\tilde{x}, x_N^*)}{\sqrt{N}}$ giving a $(1-\alpha)$-level confidence interval of $[0, Gap_N(\tilde{x}, x_N^*) + \epsilon_{Gap}]$. The candidate solution $\tilde{x}$ is again deemed optimal when the width of the confidence interval falls below some threshold, and indeed under i.i.d. sampling, the stopping criteria will be achieved as the sample size N grows large (Bayraksan and Morton, 2006). Specifically, the stopping algorithm is as follows:

Algorithm 5.2. *Single Replication Stopping Procedure (1RP)*

1. *Inputs to the algorithm are: a candidate solution $\tilde{x} \in X$, sample size N, and a value of $0 < \alpha < 1$ for a $(1-\alpha)$-level confidence interval.*

2. *Sample i.i.d. observations $\xi^1, \ldots, \xi^N$ from the distribution of ξ.*

    a. *Using this sample, solve the sampled stochastic program to obtain solution x_N^*. Be sure to store the values $G(x_N^*, \xi^i)$.*

   b. *Using this sample and the candidate solution* $\tilde{x}$*, obtain the values* $G(\tilde{x}, \xi^i)$.

3. *Calculate* $Gap_N(\tilde{x}, x_N^*)$ *and* $s_N^2(\tilde{x}, x_N^*)$ *as defined in (5.11) and (5.12) respectively.*

4. *Output a one-sided confidence interval* $[0, Gap_N(\tilde{x}, x_N^*) + \frac{t_{N-1,\alpha} s_N(\tilde{x}, x_N^*)}{\sqrt{N}}]$ *on the optimality gap* $\mu_{\tilde{x}}$.

In the same paper, two different stopping procedures using two replications are introduced. The first is the independent 2-replication procedure (I2RP). It aims to eliminate the correlation between $Gap_N(\tilde{x}, x_N^*)$ and $s_N^2(\tilde{x}, x_N^*)$ by estimating the two quantities using two separate replications. In the first replication of N i.i.d samples, the gap estimate is calculated as in 1RP. However, to estimate the sample variance, a second replication of N i.i.d. samples is drawn and the sampled stochastic program is solved with those samples to obtain a solution x_N^{**}. The sample variance $s_N^2(\tilde{x}, x_N^{**})$ is estimated by replacing all instances of x_N^* in Equation (5.12) with x_N^{**}, and it then follows that $\epsilon_{Gap} = \frac{t_{N-1,\alpha} s_N(\tilde{x}, x_N^{**})}{\sqrt{N}}$. The resulting confidence interval is also asymptotically valid.

Algorithm 5.3. *Independent 2-Replication Stopping Procedure (I2RP)*

1. *Inputs to the algorithm are: a candidate solution* $\tilde{x} \in X$*, sample size* N*, and a value of* $0 < \alpha < 1$ *for a* $(1-\alpha)$*-level confidence interval.*

2. *Sample i.i.d. observations* $\xi^1, \ldots, \xi^N$ *from the distribution of* ξ.

   a. *Using this sample, solve the sampled stochastic program to obtain solution* x_N^**. Be sure to store the values* $G(x_N^*, \xi^i)$.

   b. *Using this sample and the candidate solution* $\tilde{x}$*, obtain the values* $G(\tilde{x}, \xi^i)$.

3. *Sample an additional* N *i.i.d. observations* $\xi^{N+1}, \ldots, \xi^{2N}$ *from the distribution of* ξ.

   a. *Using this second sample, solve the sampled stochastic program to obtain solution* x_N^{**}. *Be sure to store the values* $G(x_N^{**}, \xi^i)$.

   b. *Using this second sample and the candidate solution* $\tilde{x}$, *obtain the values* $G(\tilde{x}, \xi^i)$.

4. *Calculate* $Gap_N(\tilde{x}, x_N^*)$ *and* $s_N^2(\tilde{x}, x_N^{**})$ *as described above.*

5. *Output a one-sided confidence interval* $[0, Gap_N(\tilde{x}, x_N^*) + \frac{t_{N-1,\alpha} s_N(\tilde{x}, x_N^{**})}{\sqrt{N}}]$ *on the optimality gap* $\mu_{\tilde{x}}$.

The second stopping procedure is essentially a MRP with 2 replications each with N i.i.d. samples. A gap estimate and a sample variance is calculated for each replication and are averaged across the two replications. This is called the averaged 2-replication procedure (A2RP). The confidence interval here differs slightly from the previous procedures as there are $2N$ samples rather than N. This too yields an asymptotically valid confidence interval.

Bayraksan and Morton (2007) later extend the single replication procedure to a sequential sampling procedure. Rather than drawing a new sample each time a candidate solution needs to be tested, additional sample points are appended to the current sample. This saves a great deal of computational work as it cuts down the number of times the stochastic optimization problem needs to be solved. They show that if the gap estimator has a finite moment generating function, then the $(1 - \alpha)$-level confidence interval generated under this procedure is asymptotically valid.

In Section 5.6, we show that the 1RP and I2RP stopping procedures can be modified to also yield asymptotically valid confidence intervals with non-i.i.d. sampling schemes.

5.3 Sensitivity Analysis

Sensitivity analysis is the study of how the variation in the output of a model can be explained by its input factors. For our purposes, the output of the model is the optimal value of the

second stage of a stochastic program and the inputs are the random right-hand side values or left-hand side coefficients of the constraints. There are a number of different methods for sensitivity analysis— including derivative-based methods, screening methods, sampling-based methods, and variance-based methods. In this section, we will give an overview of some of these methods and discuss their advantages and drawbacks. Further details about sensitivity analysis can be found in Saltelli, Chan, and Scott (2000). For purposes of this section, let us denote the input variables to our function as $\{\xi_j, j = 1, \ldots, s\}$ and the output as $Y = g(\xi_1, \ldots, \xi_s)$. We now give a brief overview of some of the more common methods of sensitivity analysis.

5.3.1 Local Methods

Local sensitivity analysis methods focus on how the output behaves in the neighborhood of a specific fixed point for the input variables. The most commonly used local sensitivity measure is the gradient $\nabla Y = (\frac{\partial Y}{\partial \xi_1}, \ldots, \frac{\partial Y}{\partial \xi_s})$. This measure is often used in deterministic optimization where the $\{\xi_j\}$ correspond to the right-hand sides of the constraints and are not variable at all (leaving an easy choice for the fixed point at which to calculate the gradient). In this case, the gradient for each constraint corresponds to its dual multiplier— information which is readily available from most optimization software. In stochastic optimization, however, since the right-hand sides of the constraints may be random, ∇Y will change depending on the values of ξ for each scenario. It will also change depending on the values of the first-stage decision variables. Thus using the gradient evaluated at a single point as a sensitivity measure is not so informative. Another drawback to partial derivatives is that they only measure how the output changes as one input variable changes at a time.

An improvement is to use Principal Components Analysis (PCA) to measure how the output changes as all of the input variables change simultaneously. We can sample N re-

alizations, $\xi^1, \ldots, \xi^N$, of the input vector ξ and solve stochastic program (5.2) with each realization to obtain optimal values $Y^1, \ldots, Y^N$. Let $Z_j = \frac{\partial Y}{\partial \xi_j}$ (or some function of $\frac{\partial Y}{\partial \xi_j}$) and Z be the column vector with j^{th} entry Z_j. One can create a sensitivity matrix S equal to the sample covariance matrix of the components of Z. Note that while we assume that the input random variables in stochastic program (5.2) are uncorrelated, the components of Z will likely be correlated due to the interaction of the constraints of the stochastic program. Then since S is a covariance matrix and hence positive semi-definite, it can be decomposed as $C\Lambda C^T$, where the rows C_i of matrix C are the eigenvectors of S and the matrix Λ is a diagonal matrix whose nonzero entries are the eigenvalues λ_j of S. If λ is the vector of the eigenvalues with entries λ_j, then Z can be transformed into another random vector $\Psi = C^T Z$. It is easy to see that Ψ has covariance matrix $C^T C \Lambda C^T C = \Lambda$. This transforms the correlated variables Z_j into a set of uncorrelated variables Ψ_j called *principal components*. The principal components are just linear combinations of the original random variables and the variance of the principal components are just the eigenvalues. Thus by matching the principal components to the original variables, we can determine which variables contribute most to the variance of the function. To determine the number of important variables, we renumber the eigenvalues in descending order and calculate the trace of S (the sum of all the eigenvalues). Then the number p of important variables is the smallest number p satisfying

$$\frac{\sum_{j=1}^{p} \lambda_j}{\text{trace}(S)} \geq \theta \tag{5.13}$$

for some constant $0 < \theta < 1$.

To determine *which* variables are important, we look at the spectral decomposition of S:

$$S = C\Lambda C^T = \sum_{i=1}^{s} \lambda_i C_i C_i^T. \tag{5.14}$$

Suppose that λ_1 is much larger than the other eigenvalues. Then any differences in magnitude between the entries of S are primarily determined by $C_1 C_1^T$. Since we are interested in the diagonal terms of S (the variance terms), the p largest and most important diagonal terms correspond to the p largest entries (in absolute value) of eigenvector C_1. If multiple eigenvalues are of similar magnitude, we can take a weighted average of the corresponding eigenvectors and use similar reasoning to determine the important variables.

5.3.2 Screening Methods

Screening methods are similar to the derivative-based methods described above in that they look at perturbations around a few base values; however, the perturbations tend to be larger than those used in derivative-based methods. The main goal of screening methods is to determine with low computational effort a small subset of factors that are important to the problem from some much larger group of factors.

The most basic type of screening method is the one-at-a-time (OAT) design where the sensitivity measure of each input variable is determined by perturbing the factor of interest by δ from some fixed point while keeping the other factors constant, and then dividing the change in function value by δ. Included in these designs are *standard OAT designs* where one factor at a time is varied from some standard point and *strict OAT designs* where one factor is changed from the condition of the prior experimental run. As with the derivative-based methods, these are local methods.

Morris (1991) proposed a OAT design to calculate a global sensitivity measure. In his method the number of experimental runs is proportional to the number of input factors. For each factor, he computes r local sensitivity measures at different points and then calculates the sample mean and sample standard deviation over the points. Factors whose effects have high mean have a high linear effect and are deemed important. Factors with a high standard

deviation are either highly nonlinear or interact greatly with other factors. Factors with low mean and low standard deviation are unimportant.

A broader category of screening methods are general factorial designs where each factor is set to either some high level (+) or some low level (−). A full factorial design contains all 2^s combinations of + and − for the s factors and can produce an enormous number of calculations. Fractional factorial designs (FFD) look at only a subset of these combinations with the idea that after some point, higher-level interactions become negligible. Andres and Hajas (1993) developed the *iterated fractional factorial design* (IFFD) in which the number of runs required is fewer than the number of factors. IFFD is an example of a group screening method, and it estimates the main effects, quadratic effects, and two-factor interactions of the factors. In each iteration of IFFD, the factors are randomly split into groups and each factor is assigned a random level – high, medium, or low (factors in the same group may be assigned different levels). The groupings change with each iteration. Groups with no important factors should all behave similarly regardless of the factor levels, while the presence of influential factors in a group should greatly effect the group's results. Andres (1997) later infused some ideas from Latin Hypercube sampling to improve IFFD. Namely, rather than sampling factors at three levels, we sample at K levels; and rather than having each level be a fixed value, the levels represent intervals and a random value is sampled from that interval. With multiple factors, this is analogous to randomly selecting a hypercube and then sampling a point inside that hypercube.

Another group screening technique is sequential bifurcation originally proposed by Bettonvil (1990) (see also Bettonvil and Kleijnen (1997)). Factors are again split into groups and by sampling the factors in each group at multiple levels (typically just low and high levels), we can determine whether the group as a whole is influential. If a group is declared unimportant, then all of the factors within that group are deemed unimportant and are

disregarded for the remainder of the algorithm. If the group is important, then it is split into two subgroups and the process is repeated. The process ends when all of the remaining groups contain only one element and those elements are labeled as important. In order for sequential bifurcation to work properly, the signs for all of the main effects must be known and nonnegative to prevent multiple factors in a group from canceling each other out. This of course will not always be the case.

On the whole, screening methods are computationally efficient techniques to determine a subset of most influential factors from a much larger subset of input factors. However, they have one major drawback. They qualitatively label factors as important or unimportant but do not in general provide good quantitative measures for the order of importance. In turn, these algorithms tend to have high Type I errors (false positives) as the threshold for importance may vary depending on the application. Often screening methods are used as a first step in sensitivity analysis to eliminate unimportant random variables, and then some other method is used to quantify the level of importance.

5.3.3 Sampling-Based Methods

The next class of sensitivity analysis methods are sampling-based methods. Rather than setting factors at specified levels, we sample from the input distributions of the factors. Traditionally, Monte Carlo sampling has been used, but other sampling methods such as Latin Hypercube sampling or Quasi-Monte Carlo sampling may be used instead. Importance measures are then calculated using a variety of statistical techniques.

Perhaps the simplest technique for sensitivity analysis is to generate scatterplots. We plot each input variable against the corresponding output variable and visually analyze the relationship between them. While an effective technique, it is unfortunately not practical in an automated algorithm.

Regression analysis (for reference, see Tamhane and Dunlop (2000)) is also commonly used in sensitivity analysis. In regression analysis, the output variable Y^i from sample i can be estimated as a linear function $\hat{Y}^i = \hat{\beta}_0 + \sum_{j=1}^{s} \hat{\beta}_j \xi_j^i$ of the input variables $\xi_1, \ldots, \xi_s$. The quality of the regression is measured by

$$R^2 = \frac{\sum_{i=1}^{N}(Y^i - \hat{Y}^i)}{\sum_{i=1}^{N}(Y^i - \bar{Y})}$$

where $\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y^i$. The regression coefficients $\{\hat{\beta}_j, j = 1, \ldots, s\}$ can then serve as importance measures for the random variables.

From the regression, we can also calculate the standardized regression coefficients (SRC) $\hat{r}_{\xi_j,Y} = \frac{\hat{\beta}_j \hat{s}_j}{\hat{s}}$ where $\hat{s}_j$ is the sample standard deviation of ξ_j and $\hat{s}$ is the sample standard deviation of Y. These are the regression coefficients if the input and output variables of the regression were all normalized to have mean 0 and standard deviation 1. SRC's are equivalent to the Pearson correlation coefficients (PEAR)

$$\hat{r}_{\xi_j,Y} = \frac{\sum_{i=1}^{N}(\xi_j^i - \bar{\xi}_j)(Y^i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(\xi_j^i - \bar{\xi}_j)^2}\sqrt{\sum_{i=1}^{N}(Y^i - \bar{Y})^2}}. \tag{5.15}$$

When the input variables $\xi_j, j = 1, \ldots, s$ are independent, $r_{\xi_j,Y}$ can be used as an importance measure. A related measure is the partial correlation coefficients (PCC) which provide a measure of the linear relationship between ξ_j and Y after the linear effects of the other random variables have been removed. Regression analysis and correlations can further be combined with other statistical tests, such as hypothesis tests which can determine whether $\hat{\beta}_j = 0$ for factor j.

Another regression technique is stepwise regression where we begin by fitting a regression model to the single input variable that most impacts the uncertainty of the output variable. In the next iteration, the second-most influential input variable is added to the model and

regression coefficients are calculated for this two-variable model. This process continues until adding an additional input variable has a minimal impact to the R^2 of the model. The final regression model contains only the important input variables.

While the models above improve upon screening methods in that they give quantitative measures of importance, they too still have one major drawback. The importance measures are only as good as the underlying regression. This is especially problematic for highly nonlinear models. Rank transformations, where the random variables are replaced by their respective ranks (1=smallest, etc.), can increase the R^2 of the regression, but still require monotonic relationships between the input and output variables to be truly effective. Similarly, Pearson correlation and partial correlation can also be calculated using ranks but with the same monotonicity requirements.

5.3.4 Variance-Based Methods

Variance-based sensitivity analysis methods assume that the importance of each input variable can be determined solely by measuring its contribution to the variance of the output variable.

One way to measure the effect of input variable ξ_j on the variance of Y is to determine how much that total variance would be reduced if we fixed the value of ξ_j, i.e.,

$$\frac{\mathrm{Var}_{\xi_j}(\mathbb{E}[Y|\xi_j = k])}{\mathrm{Var}(Y)}. \tag{5.16}$$

McKay (1995) named this ratio the *correlation ratio* and the numerator the *variance correlation expectation (VCE)*. Similarly Hora and Iman (1986) previously used $\sqrt{\mathrm{VCE}_j}$ as a measure of importance, where VCE_j is the numerator of (5.16) for factor ξ_j. Unfortunately, while mathematically correct, these methods were found to lack robustness and improve-

ments have been suggested such as rank transformations (Homma and Saltelli, 1994) and replacing Y with $\log(Y)$ (Iman and Hora, 1999).

Sobol (1993) proposed a method of determining importance measures using the ANOVA decomposition of the function. Recall from Section 2.3.2 that a function $g(\xi_1, \ldots, \xi_s)$ can be decomposed into a sum of orthogonal functions each corresponding to a different subset of the input variables. This decomposition has the property that the total variance of the function (denoted σ^2) is equal to the sum of the variances of each of the component functions

$$\sigma^2 = \sum_{A \subseteq \{1,\ldots,s\}} \sigma_A^2.$$

Sobol originally based the decomposition on multiple Fourier Haar series, but later represented the decomposition using multiple integrals, which are typically calculated using Monte Carlo (or QMC) integration. Using the decomposition, he defines two sensitivity measures for each random variable. The *first-order sensitivity index* of ξ_j (sometimes referred to as the *main effect*) is

$$S_j = \frac{\sigma_j^2}{\sigma^2} \tag{5.17}$$

and the *total sensitivity index (TSI)* is

$$\mathrm{TS}_j = \frac{\sum_{\{A \subseteq \{1,\ldots,s\} : j \in A\}} \sigma_A^2}{\sigma^2}. \tag{5.18}$$

Note that the numerator of (5.18) sums up the variance terms for all of the subsets of random variables that contain variable ξ_j.

Homma and Saltelli (1996) devised a method to compute the total sensitivity indices without calculating the variance terms for every component. To calculate TS_j they partition the random variables into two subsets – one containing only random variable ξ_j and the

other with the remaining variables. Using a generalized ANOVA decomposition, the variance can be decomposed as $\sigma^2 = \sigma_j^2 + \sigma_{\sim j}^2$ and the total sensitivity index can be calculated as $\text{TS}_j = 1 - S_{\sim j}$. While this eliminates some of the calculations involved in Sobol's method, the resulting method is still significantly more computationally expensive to run and more difficult to code than the screening and sampling-based sensitivity methods.

A final variance-based method is the *Fourier amplitude sensitivity test (FAST)* method developed by Cukier, Fortuin, Shuler, Petschek, and Schaibly (1973). It is an alternative approach to calculate the same indices as the Sobol method. The main idea behind this method is to convert the s-dimensional integral of $Y = g(\xi_1, \ldots, \xi_s)$ into a one-dimensional integral over some scalar variable $t \in (-\pi, \pi)$ by using the transformations

$$\xi_j = G_j(\sin(\omega_j t))$$

where $\{G_j, j = 1, \ldots, s\}$ are transformation functions and $\{\omega_j\}$ are integer angular frequencies. First-order sensitivity indices can be calculated for each variable by using the Fourier decomposition of $\mathbb{E}[Y]$ and $\text{Var}(Y)$. It turns out that these first-order indices are identical to those calculated using the Sobol method. Saltelli, Tarantola, and Chan (1999) proposed a method called *Extended FAST* to compute the total sensitivity indices under FAST. FAST and Extended FAST are computationally more efficient than the Sobol method, but still less so than the other types of sensitivity analysis methods.

5.4 Identifying the Set of Important Variables

The first task in a padded sampling scheme is identifying the set of important random variables that will be estimated using QMC in the sampling scheme. Since Quasi-Monte Carlo sampling typically results in a lower integration error than Monte Carlo sampling,

it seems natural to label the important variables as the ones that contribute the most to the overall variance of the function. For the two-stage stochastic program, the function of interest is the second stage optimal value $Q(x, h, T)$. Using traditional sensitivity analysis methods to determine the importance of the random variables is problematic for a number of reasons. First, the random quantities h and T are in the right-hand sides of the constraints of the optimization problem. This can be remedied by writing the dual of the second stage stochastic program:

$$\sup\{\pi^T(h(\xi) - T(\xi)x) : \pi^T W \leq q^T, \pi \geq 0\} \tag{5.19}$$

Thus the total variance of the optimal value is

$$\text{Var}\left(\sum_k \pi_k^*(h_k - \sum_j T_{kj}x_j)\right) \tag{5.20}$$

where π_k^* is the optimal dual multiplier for constraint k. However, the optimal dual multipliers in the function depend on the values of the random variables (right-hand sides of the constraints) and are often correlated with each other, due to the interacting structure of those constraints. This makes evaluating the second stage optimal value a nontrivial task, and thus sensitivity measures involving a large number of function evaluations are unattractive. Further, when there is interaction among the constraints, $Q(x, h, T)$ is a nonlinear function with respect to the random variables. This prevents us from using most regression-based methods. Finally, $Q(x, h, T)$ additionally depends on the vector x of stage 1 variables which is not random. Thus, it is not only necessary to determine some global sensitivity measure for h and T, sensitivity measures must be calculated for different values of x. Solving the first-stage stochastic program is a particularly computationally intensive task.

Due to the structure of our problem and the need to limit the number of stochastic pro-

grams we solve, we have developed our own heuristics for determining the importance of each random variable — i.e., its contribution to the overall variance of the optimal value. Assuming that the primal problem has m constraints and n variables, the summation in Equation (5.20) has $M := m + mn$ terms. To simplify notation, we can rewrite the optimal value as $\sum_{i=1}^{M} Z_i$, where the Z_i are defined accordingly. Note that in Equation (5.20), some of the h_k and T_{kj} terms may not be random. Let $s \leq M$ be the number of random components among the h_k and T_{kj}. Each Z_i contains at most one of these random components. Without loss of generality, assume that the Z_i are labeled in such a way that these random components correspond to $Z_1, \ldots, Z_s$. Note that while we assume that the random components themselves are mutually independent, the Z_i terms in the objective function are usually dependent due to the interactions of the dual multipliers. The variance of the optimal value is now

$$Var(\sum_{i=1}^{M} Z_i) = \sum_{i,j} \mathrm{Cov}(Z_i, Z_j). \tag{5.21}$$

Similarly, let V_k, $k = 1, \ldots, s$ be the portion of the total variance explained by random variable ξ_k. Then the variance of the optimal value can also be written as $\sum_{k=1}^{s} V_k$. Our goal then is to estimate the individual covariance terms in (5.21) and assign each to one of the V_k.

5.4.1 Calculating Importance Measures

We now present three heuristics for estimating the importance measures $V_k, k = 1, \ldots s$. Since the V_k represent the contributions of each random variable to the overall variance, larger values of V_k should be associated with more important random variables.

The first heuristic uses an idea similar to the first-order sensitivity index from sensitivity analysis.

Heuristic 1:

$$\hat{V}_k = \mathrm{Var}(Z_k), \quad k = 1, \ldots, s. \qquad \square \tag{5.22}$$

This is a very crude approximation which completely ignores any dependence between the dual multipliers. It is analogous to a main effect term from sensitivity analysis. Its advantage is that it only involves estimating at most s terms of the covariance matrix so it may be preferable for large problems.

Our second heuristic attempts to apportion the covariance terms to the input variables.

Heuristic 2:

$$\hat{V}_k = \left| \mathrm{Var}(Z_k) + \sum_{j=1, j\neq k}^{s} \mathrm{Cov}(Z_k, Z_j) + \sum_{j=s+1}^{M} \mathrm{Cov}(Z_j, Z_k) \right|, \quad k = 1, \ldots, s. \qquad \square \tag{5.23}$$

Here the contribution of each random variable to the overall variance includes the variance of its own Z term plus the covariance with the Z terms of the other random variables plus *both* covariance terms with the Z terms that do not have a random variable (as the second covariance term would otherwise be unassigned). We ignore covariance terms of the form $\{\mathrm{Cov}(Z_i, Z_j) : i > s, j > s\}$ since these terms should theoretically be small as there are no random variables present. Finally, we take the absolute value to account for random variables that greatly affect the variance in either direction and to adjust for computer rounding error.

This method involves estimating at most $s(M - s)$ terms in the covariance matrix (due to the symmetry of the matrix) though this number can be reduced by ignoring all of the Z terms that are identically zero. Still, since M is at least as large as the number of constraints in the primal problem, these estimates can be quite cumbersome for very large problems.

The third heuristic uses principal components analysis (PCA) on the covariance matrix to calculate importance measures. The eigenvalues and corresponding eigenvectors of the covariance matrix are calculated. We look at the eigenvector associated with the largest

eigenvalue. Then importance measure $\hat{V}_k$ is equal to the k^{th} component of that eigenvector.

5.4.2 Choosing the Number of Important Variables

Once we have calculated the $\hat{V}_k$ via one of the heuristics, we can select the subset of important random variables. It is critical to decide the cutoff point for inclusion into this subset. Due to the poor performance of pure QMC sampling in high dimensions, we cap the number of important random variables at some numbers D_{cap}. Typically, important subsets are chosen using the effective dimension in the truncation sense (2.9), i.e., by choosing a subset of random variables that accounts for some specified percentage of the total variance. Unfortunately, for some of our stochastic programs and some of our heuristics, the number of "important" variables under this method always exceeds the cap of ten random variables. So instead we have chosen to use the PCA method of determining the number of important variables for all three heuristics, even though for heuristics 1 and 2 the importance measure is not calculated using PCA. In PCA, the number of important variables is determined by the eigenvalues $\lambda_1, \ldots, \lambda_M$ of the covariance matrix S. We can reorder the eigenvalues in descending order and calculate the trace of Σ (the sum of the eigenvalues). The number of important variables from the PCA is then the number p such that $\frac{\sum_{i=1}^{p} \lambda_i}{\text{trace}(S)} \geq \theta$. For our algorithm, we assume $\theta = 0.9$. Then we use $d = \min(p, D_{cap})$ as the number of important variables in our algorithm.

5.4.3 Other Considerations

There are two other items to consider when choosing the important subset and eventually executing the padded sampling scheme. First, variance reduction is maximized for a QMC point set when the variables are in descending order of importance (Fox, 2000). Thus, our important subset should be an ordered subset to maximize the efficiency of the sampling

algorithm. Second, we add an extra rule when choosing important variables for a PLHS scheme as opposed to a PMC scheme. In PLHS, important variables are sampled using a QMC point set and the remaining variables using a Latin Hypercube point set. However, one of the properties of LHS is that the asymptotic variance is equal to just the variance of the nonlinear terms of the ANOVA decomposition. Thus, it should be most efficient to assign random variables whose effects are mostly linear to the Latin Hypercube point set regardless of that variable's importance measure. A random variable whose effect is mostly linear should not have significant interactions with the other random variables. Heuristic 2 accounts for the non-diagonal terms in the covariance matrix (i.e., the terms measuring interactions between random variables) while Heuristic 1 does not. The two heuristics should give similar importance measures for random variables whose interactions with other random variables are insignificant. Thus, we use the ratio $\frac{\hat{V}_k^{Heur1}}{\hat{V}_k^{Heur2}}$ to measure the linear effect. This involves the following steps:

- Calculate each random variable's importance measure from Heuristics 1 and 2. For the k^{th} random variable, call these $\hat{V}_k^{Heur1}$ and $\hat{V}_k^{Heur2}$ respectively.
- If $0.9 < \frac{\hat{V}_k^{Heur1}}{\hat{V}_k^{Heur2}} < 1.1$, then the random variable has a mostly linear effect and should be sampled using LHS.

In summation, we propose the following algorithm to select the important subset:

Algorithm 5.4. *Selecting the Important Set I*

1. *Rewrite the second stage dual objective as* $\sum_k Z_k$ *as described earlier in this section.*
2. *Calculate importance measures* $\{\hat{V}_k, k = 1, \ldots, s\}$ *using one of the three heuristics. For PLHS, it will additionally be necessary to calculate the importance measures from heuristics 1 and 2 regardless of the heuristic used.*

3. *Let the important set* $I = \emptyset$*, and unimportant set* $U = \{1, \ldots, S\}$.

4. *Rank the* $\hat{V}_k$ *in descending order.*

5. *Calculate the number of important variables* p *using the eigenvalues of the covariance matrix. Then let* $d = \min(p, 10)$ *be the number of important variables for the algorithm.*

6. *Select the largest remaining* $\hat{V}_k$*.*

   a. *For PLHS sampling, calculate the ratio* $\frac{V_k^{Heur1}}{V_k^{Heur2}}$*. If this ratio is between* 0.9 *and* 1.1*, then set* $V_k = 0$*, and repeat step 5.*

   b. *If* $\frac{V_k^{Heur1}}{V_k^{Heur2}}$ *is not between* 0.9 *and* 1.1 *in step 5a or for PMC sampling, let* $I \leftarrow I \bigcup \{k\}$ *and* $U \leftarrow U - \{k\}$.

7. *If* $|I| = d$ *then terminate the algorithm with important set* I*. Otherwise, return to step 5.*

We are now ready to implement this method into an external sampling algorithm to solve two-stage stochastic programs.

5.5 The Algorithm

All of our heuristics in the prior section require the calculation of a covariance matrix from the second stage stochastic program (5.2). However, in order to solve the second stage program, we need a first stage candidate solution $\tilde{x}$. Thus the algorithm we propose is an iterative algorithm where the sampled stochastic program is solved to obtain a new $\tilde{x}$ at each iteration using a padded sample and then that $\tilde{x}$ is used to determine the set of important variables for the next iteration. The initial $\tilde{x}$ value is determined by solving the sampled optimization problem using a Monte Carlo point set (as we have not yet identified the important random

variables by that time). To estimate each random variable's contribution $\hat{V}_k$ to the variance and to determine the set of important random variables, we estimate $\mathbb{E}[Q(x,h,T)]$ using a Monte Carlo point set. Then from the samples we can estimate the covariance matrix of the Z_k terms (as defined in section 5.4). In order to gauge each random variable's full effect on the overall variance, it is essential that we employ a sampling scheme with no variance reduction while estimating the covariance terms. The algorithm terminates via a $1RP$ or $2RP$ stopping criterion when the ratio of the width $W$of the 95%-confidence interval on the optimality gap at $\tilde{x}$ is less than some tolerance ε. We detail the full External Sampling Algorithm with Padded Sampling (ES-PAD) below. Specifically when the padded sampling scheme is padded Monte Carlo, we call the algorithm ES-PMC, and when it is padded Latin Hypercube we call it ES-PLHS.

Algorithm 5.5. *An External Sampling Algorithm with Padded Sampling (ES-PAD)*

1. Initialization:

   (a) *Set $\varepsilon > 0$ for the stopping criterion. Select initial sample size n_0 and sample increase multiplier κ.*

   (b) *Set iteration count $i \leftarrow 0$, important set $I^{(0)} = \emptyset$, let sample size $N^{(0)} = n_0$.*

   (c) *Using a Monte Carlo point set of size $N^{(0)}$, solve the sampled optimization program to determine first stage candidate solution $\tilde{x}^{(0)}$ and optimal value $\psi^{(0)}$.*

2. Increment: *Let $i \leftarrow i+1$ and $N^{(i)} = \kappa N^{(i-1)}$.*

3. *Using $N^{(i)}$* Monte Carlo *samples to solve the second stage stochastic program at candidate solution $\tilde{x} = \tilde{x}^{(i-1)}$, calculate the estimates $\hat{V}_k$, $k = 1, \ldots, s$ using one of the three heuristics described in section 5.4.*

4. *Using Algorithm 5.4, select the important set $I^{(i)}$ for iteration i.*

5. *Using a padded point set— i.e., a Quasi-Monte Carlo point set for random variables $k \in I^{(i)}$ and a padded (Monte Carlo or LHS) point set on random variables $k \notin I^{(i)}$ — solve the sampled optimization program to determine first stage candidate solution $\tilde{x}^{(i)}$ and optimal value $\psi^{(i)}$.*

6. *Calculate the width $W^{(i)}$ of the confidence interval on the optimality gap at the candidate solution $\tilde{x}^{(i)}$ using one of the stopping criteria (1RP or I2RP). If $\frac{W^{(i)}}{\psi^{(i)}} \leq \varepsilon$ then terminate the algorithm with optimal first stage solution $\tilde{x}^{(i)}$ and optimal value $\psi^{(i)}$. Otherwise return to step 2.* □

We will compare this algorithm to the corresponding algorithm ES-NoPAD where all sampling is done with a single unpadded sampling scheme, thus making it unnecessary to estimate the importance of the random variables. When that unpadded sampling scheme is crude Monte Carlo, we call the algorithm ES-MC. Latin Hypercube will be called ES-LHS and pure QMC will be ES-QMC.

Algorithm 5.6. *An External Sampling Algorithm with No Padding (ES-NoPAD)*

1. Initialization:

   (a) *Set $\varepsilon > 0$ for the stopping criterion. Select initial sample size n_0 and sample increase multiplier κ.*

   (b) *Set iteration count $i \leftarrow 0$, important set $I^{(0)} = \emptyset$, let sample size $N^{(0)} = n_0$.*

   (c) *Using an unpadded point set of size $N^{(0)}$, solve the sampled optimization program to determine first stage candidate solution $\tilde{x}^{(0)}$ and optimal value $\psi^{(0)}$.*

2. Increment: *Let $i \leftarrow i + 1$ and $N^{(i)} = \kappa N^{(i-1)}$.*

3. *Using an unpadded point set, solve the sampled optimization program to determine first stage candidate solution $\tilde{x}^{(i)}$ and optimal value $\psi^{(i)}$.*

4. *Calculate the width $W^{(i)}$ of the confidence interval on the optimality gap at the candidate solution $\tilde{x}^{(i)}$ using one of the stopping criteria (1RP or I2RP). If $\frac{W^{(i)}}{\psi^{(i)}} \leq \varepsilon$ then terminate the algorithm with optimal first stage solution $\tilde{x}^{(i)}$ and optimal value $\psi^{(i)}$. Otherwise return to step 2.* □

In Section 5.7, we will present numerical results of our five algorithms (ES-PMC, ES-PLHS, ES-MC, ES-LHS, and ES-QMC) run with the different importance heuristics and stopping criteria.

5.6 A Stopping Criterion for the Algorithm

The stopping procedures in Section 5.2.2 all assume that the samples within each replication are i.i.d. Since the sampling methods in our algorithms are mostly non-i.i.d. (except for ES-MC), we would like to take advantage of the presumably lower sample variance these sampling methods offer so that we may reach the stopping criterion after fewer samples. For our algorithm, we will use only the 1RP (Algorithm 5.2) and I2RP (Algorithm 5.3) stopping procedures. Our amended stopping procedures will work the same as the original procedures except with Monte Carlo sampling replaced by a non-i.i.d. sampling scheme to estimate the mean and variance of the optimality gap $\mu_{\tilde{x}}$ at candidate solution $\tilde{x}$. Since the asymptotic variance under any of LHS, PMC, and PLHS is no worse than the asymptotic variance under Monte Carlo sampling (see Equations 4.3, 4.2, and 4.1, respectively), this should lead to a tighter confidence interval around the optimality gap. With this change though, it is necessary to show that the confidence intervals for these amended stopping procedures are asymptotically valid. We will show below that this result is true for PLHS, PMC, and LHS for both the 1RP and I2RP stopping procedures. We will specifically show the proof for the 1RP sampling procedure for PLHS. The proofs for the other five combinations follow the

same steps substituting the appropriate sampling method and stopping procedure.

For reference, we state without proof a lemma and a theorem regarding stochastic programs (1.1) and (1.2) from Rubinstein and Shapiro (1993) that will be needed in the proofs of this section. These results assume:

Assumption 5.7.

1. *For almost every ξ, the function $G(\cdot, \xi)$ is continuous, and*
2. *The family $\{|G(x, \xi)|, x \in X\}$ is dominated by an integrable function.*

Note that Assumption 5.7 will hold if we have Assumption 5.1.

Lemma 5.8. *Suppose Assumption 5.7 holds and let $g(x) = \mathbb{E}[G(x, \xi)]$. Then the expected value function $\hat{g}_N(x) = \frac{1}{N}\sum_{i=1}^{N} G(x, \xi^i)$ is continuous on X. If, in addition, the set X is compact, then w.p.1. $\hat{g}_N(x)$ converges to $g(x)$ uniformly on X.*

Proof: See Rubinstein and Shapiro (1993). □

Theorem 5.9. *Suppose Assumption 5.7 holds and that X is compact. Let $v^* = \min_{x \in X} g(x)$ and $\hat{v}_N = \min_{x \in X} \hat{g}_N(x)$. Then $\hat{v}_N$ converges to v^* w.p.1. Moreover, if x^* is a unique minimizer of $g(x)$ over X, then $\hat{x}_N$ converges to x^* w.p.1.*

Proof: See Rubinstein and Shapiro (1993). □

With i.i.d. sampling, Bayraksan and Morton (2006) show the asymptotic validity of the confidence interval for the optimality gap at candidate solution $\tilde{x}$ for the 1RP stopping algorithm under Assumption 5.1, or, more formally, that

$$\liminf_{N \to \infty} \mathbb{P}\left(\mu_{\tilde{x}} \leq Gap_N(\tilde{x}, x_N^*) + \frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sqrt{N}}\right) \geq 1 - \alpha.$$

Similarly, the confidence interval for I2RP replaces the $s_N(\tilde{x}, x_N^*)$ term with $s_N(\tilde{x}, x_N^{**})$.

Although the sampling methods we wish to use are no longer i.i.d., they are still unbiased, and thus, the results from Rubinstein and Shapiro (1993) will still apply. The following proofs will mirror those of Bayraksan and Morton (2006) with comments noting the appropriate changes to account for our non-i.i.d. sampling method. While we only give the proofs for the 1RP stopping procedure under PLHS, the proofs for PMC and LHS, as well as the corresponding I2RP proofs, will follow exactly the same steps and reasoning.

We begin with some notation. Let $\hat{g}_N(x)$ be defined as in Lemma 5.8; let

$$\sigma^2(\tilde{x}, x^*) = \mathrm{Var}(G(\tilde{x}, \xi) - G(x^*, \xi))$$

be the true variance of the optimality gap; and let

$$s_N^2(\tilde{x}, x^*) = \frac{1}{N-1} \sum_{i=1}^{N} ((G(\tilde{x}, \xi^i) - G(x^*, \xi^i)) - (\hat{g}_N(\tilde{x}) - \hat{g}_N(x^*)))^2$$

be the sampled variance of the optimality gap with N samples under PLHS.

Proposition 5.10. *Assume Assumption 5.1 holds, $\tilde{x} \in X$, and that $\xi^1, \ldots, \xi^N$ are PLHS samples of ξ. Let $X^* = \mathrm{argmin}_{x \in X} \mathbb{E}[G(x, \xi)]$ be the set of optimal solutions to the stochastic program. Then,*

(i) $\hat{v}_N$ converges to v^*

(ii) All limit points of $\{\hat{x}_N\}$ lie in X^*, w.p.1.

Proof: Since Theorem 5.9 also holds for PLHS, this proof will be identical to the first part of Proposition 1 of Bayraksan and Morton (2006). □

Proposition 5.11. *Assume Assumption 5.1 holds, $\tilde{x} \in X$, and that $\xi^1, \ldots, \xi^N$ are PLHS samples of ξ. Let $X^* = \mathrm{argmin}_{x \in X} \mathbb{E}[G(x, \xi)]$ be the set of optimal solutions to the stochastic*

program, and let $x^*_{min} \in argmin_{x \in X^*} \mathrm{Var}[G(\tilde{x}, \xi) - G(x, \xi)]$ *and* $x^*_{max} \in argmax_{x \in X^*} \mathrm{Var}[G(\tilde{x}, \xi) - G(x, \xi)]$ *be the optimal solutions with minimum and maximum variance, respectively. Then,*

$$\sigma^2(\tilde{x}, x^*_{min}) \leq \liminf_{N \to \infty} s^2_N(\tilde{x}, \hat{x}_N) \leq \limsup_{N \to \infty} s^2_N(\tilde{x}, \hat{x}_N) \leq \sigma^2(\tilde{x}, x^*_{max})$$

Proof: To prove this, we need to show that the sequence of continuous functions $s^2_N(\tilde{x}, \hat{x}_N)$ converges to $\sigma^2(\tilde{x}, x^*)$ uniformly, w.p.1. on X. By invoking Lemma 5.8, which also works on our padded sampling schemes, the rest of this proof will follow that of Proposition 1 of Bayraksan and Morton (2006). □

For the following theorem, we must introduce one additional piece of notation:

$$\sigma^2_{PLHS}(\tilde{x}, x^*) = \mathrm{Var}_{PLHS}(G(\tilde{x}, \xi) - G(x^*, \xi))$$

is the asymptotic PLHS variance of the optimality gap (refer to Equation (4.1) for an expression for the PLHS asymptotic variance). Please note that the basic structure of the proof below follows that of Theorem 2 of Bayraksan and Morton (2006) with the necessary modifications for padded sampling and some minor modifications for notation. For completeness in explaining the effect of the padded sampling, the entire proof is included. Also note that in the PLHS case, the result holds solely due to the PLHS central limit theorem proven in Chapter 4 of this thesis.

Theorem 5.12. *Assume Assumption 5.1 holds,* $\tilde{x} \in X$*, and that* $\xi^1, \ldots, \xi^N$ *are from a PLHS sample of* ξ*. Then given* $0 < \alpha < 1$ *for the 1RP,*

$$\liminf_{N \to \infty} \mathbb{P}\left(\mu_{\tilde{x}} \leq Gap_N(\tilde{x}, x^*_N) + \frac{z_\alpha s_N(\tilde{x}, x^*_N)}{\sqrt{N}}\right) \geq 1 - \alpha. \tag{5.24}$$

Proof:

When $\tilde{x} \in X^*$, Equation (5.24) holds trivially as $\mu_{\tilde{x}} = 0$. So, suppose $\tilde{x} \notin X^*$. Then, since $\hat{v}_N = \min_{x \in X} \hat{g}_N(x)$, we have that

$$Gap_N(\tilde{x}, x_N^*) = \hat{g}_N(\tilde{x}) - \hat{v}_N \geq \hat{g}_N(\tilde{x}) - \hat{g}_N(x), \quad \forall x \in X.$$

Replacing x by $x_{min}^* \in \text{argmin}_{x \in X^*} \text{Var}[G(\tilde{x}, \xi) - G(x, \xi)]$ we obtain,

$$\begin{aligned}
& \mathbb{P}\left(Gap_N(\tilde{x}, x_N^*) + \frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sqrt{N}} \geq \mu_{\tilde{x}} \right) \\
\geq \; & \mathbb{P}\left(\hat{g}_N(\tilde{x}) - \hat{g}_N(x_{min}^*) + \frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sqrt{N}} \geq \mu_{\tilde{x}} \right) && (5.25) \\
= \; & \mathbb{P}\left(\frac{(\hat{g}_N(\tilde{x}) - \hat{g}_N(x_{min}^*)) - \mu_{\tilde{x}}}{\sigma_{PLHS}(\tilde{x}, x_{min}^*)/\sqrt{N}} \geq -\frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sigma_{PLHS}(\tilde{x}, x_{min}^*)} \right) && (5.26) \\
\geq \; & \mathbb{P}\left(\frac{(\hat{g}_N(\tilde{x}) - \hat{g}_N(x_{min}^*)) - \mu_{\tilde{x}}}{\sigma_{PLHS}(\tilde{x}, x_{min}^*)/\sqrt{N}} \geq -\frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sigma(\tilde{x}, x_{min}^*)} \right) && (5.27)
\end{aligned}$$

where we assume that $\sigma_{PLHS}(\tilde{x}, x_{min}^*) > 0$. If $\sigma_{PLHS}(\tilde{x}, x_{min}^*) = 0$ then $\text{Var}[G(\tilde{x}, \xi) - G(x_{min}^*, \xi)] = \frac{1}{N}\sigma_{PLHS}(\tilde{x}, x_{min}^*) = 0$, and it would follow from (5.25) that (5.24) holds trivially. (5.27) holds because $\sigma_{PLHS}(\tilde{x}, x_{min}^*) \leq \sigma(\tilde{x}, x_{min}^*)$ and implies that the probability of being in the confidence interval (or at least its lower bound) is greater for PLHS than for i.i.d. sampling.

Let $D_N = \frac{(\hat{g}_N(\tilde{x}) - \hat{g}_N(x_{min}^*)) - \mu_{\tilde{x}}}{\sigma_{PLHS}(\tilde{x}, x_{min}^*)/\sqrt{N}}$, $a_N = -\frac{s_N(\tilde{x}, x_N^*)}{\sigma(\tilde{x}, x_{min}^*)}$, and $0 < \epsilon < 1$, and for the moment assume $\alpha \leq 0.5$ so that $z_\alpha \geq 0$. Then (5.27) can be rewritten as

$$\begin{aligned}
& \mathbb{P}(D_N \geq -z_\alpha a_N) \\
& \geq \; \mathbb{P}(D_N \geq -(1-\epsilon)z_\alpha, a_N \geq 1 - \epsilon) \\
& = \; \mathbb{P}(D_N \geq -(1-\epsilon)z_\alpha) + \mathbb{P}(a_N \geq 1 - \epsilon) - \mathbb{P}(\{D_N \geq -(1-\epsilon)z_\alpha\} \cup \{a_N \geq 1 - \epsilon\}).
\end{aligned} \tag{5.28}$$

Taking limits we obtain,

$$\liminf_{N\to\infty} \mathbb{P}\left(\mu_{\tilde{x}} \leq Gap_N(\tilde{x}, x_N^*) + \frac{z_\alpha s_N(\tilde{x}, x_N^*)}{\sqrt{N}}\right) \geq \Phi((1-\epsilon)z_\alpha),$$

where Φ denotes the distribution function of the standard normal. By Proposition 5.11, the last two terms in (5.28) both converge to 1 and cancel out. Since $\hat{g}_N(\tilde{x}) - \hat{g}_N(x_{min}^*)$ is a sample mean of PLHS random variables, by the central limit theorem in (4.5) the first term in (5.28) converges to $\Phi((1-\epsilon)z_\alpha)$. Letting ϵ shrink to zero gives the desired result, provided $\alpha \leq 0.5$. When $\alpha > 0.5$ we replace x_{min}^* by $x_{max}^* \in \text{argmax}_{x\in X^*}\text{Var}[G(\tilde{x},\xi) - G(x,\xi)]$ in (5.27) and then follow a similar argument as above. □

This proof remains the same for the I2RP when we redefine $s_N(\tilde{x}, x_N^*)$ as $s_N(\tilde{x}, x_N^{**})$ in (5.24). Also, since LHS and PMC both satisfy central limit theorems and since their asymptotic variances are both less than or equal to the asymptotic variance under i.i.d. (i.e., the true variance), the same proofs for the 1RP and I2RP stopping procedures will hold for those sampling schemes.

5.7 Numerical Results

We tested our algorithms ES-PAD (Algorithm 5.5) and ES-NoPAD (Algorithm 5.6) on four different test problems. Specifically, we tested two versions of ES-PAD — padding with Monte Carlo Sampling (ES-PMC) and padding with Latin Hypercube sampling (ES-PLHS) — and three versions of ES-NoPad — Monte Carlo sampling (ES-MC), Latin Hypercube sampling (ES-LHS), and Quasi-Monte Carlo sampling (ES-QMC) on each problem. For the padded sequences, we must determine the dimension d of the QMC point set as described in Section 5.4. For those QMC point sets as well as the point sets for ES-QMC, we used randomized (t, d)-sequences in base 2 as described by Owen (1995).

For each algorithm we tested two different stopping criteria — 1RP and I2RP. Finally, for the two padded algorithms, we tested the three heuristics described in section 5.4 for selecting the subset of important random variables used in the QMC point set. Note that it is not necessary to run these heuristics on any of the ES-NoPAD algorithms as all of the random variables are sampled using the same sampling method. All of the point sets for the random variables were generated using the publicly available random sampling routines from Friedel and Keller (2002) which we built into the SUTIL library (Czyzyk, Linderoth, and Shen 2005). The resulting stage 1 and stage 2 stochastic programs were solved using the ATR solver (Linderoth and Wright 2001). Before presenting our numerical results, we first describe in detail our four test problems.

5.7.1 Test Problems

Our four test problems were:

1. *gbd* – the airline fleet assignment problem first proposed by Dantzig (1963)
2. *LandS* – an electrical investment planning problem that originally appeared in Louveaux and Smeers (1988)
3. *apl1p* – a model for electric power capacity expansion from Infanger (1992)
4. *20term* – a vehicle allocation problem described in Bailey et al. (1999)

The first three test problems are relatively small (3 to 5 random variables) and we will explicitly write out the primal and dual problems in addition to describing the problems to provide the reader some insight on the problems. Also, given the small size of these problems, we can more easily estimate (or in some cases calculate exactly) the level of importance of each random variable. *20term* is a larger problems with 40 random variables, so we will simply describe this problem without writing out its formulation explicitly.

gbd

We begin with the fleet assignment problem first described by Dantzig (1963). Four types of aircrafts are to be assigned to five different routes (with three of the aircraft/route combinations not valid). The random variables are the customer demands for each route. This is a relatively simple problem where the importance of each random variable can be quantified in closed form. Notation for the problem is as follows:

Sets: $I = \{1, 2, 3, 4\} :=$ Set of aircrafts

$J = \{1, 2, 3, 4, 5\} :=$ Set of routes

Variables: $x_{ij} :=$ Number of aircraft of type i assigned to route j

(Combinations $(i, j) = (2, 1), (3, 1), (3, 3)$ are not valid)

$y_j :=$ Number of bumped passengers on route j

$z_j :=$ Number of empty seats on route j

Parameters: $a_i :=$ Number of available aircraft of type i

$c_{ij} :=$ Cost (in thousands per month) of assigning aircraft of type i to route j

$p_{ij} :=$ Passenger capacity (per month) for aircraft of type i on route j

$e_j :=$ Cost per bumped passengers on route j (refund of ticket cost)

$d_j :=$ Customer demand (per month) on route j (random)

The problem is then:

$$\begin{aligned}
\text{Minimize } & \sum_{i=1}^{4}\sum_{j=1}^{5} c_{ij}x_{ij} + \sum_{j=1}^{5} e_j y_j \\
\text{subject to: } & \sum_{j=1}^{5} x_{ij} \leq a_i, \quad \forall i \in I \\
& \sum_{i=1}^{4} p_{ij}x_{ij} + y_j - z_j = d_j, \quad \forall j \in J \\
& x_{21}, x_{31}, x_{33} = 0 \\
& x_{ij}, y_j, z_j \geq 0 \quad \forall i \in I, \forall j \in J
\end{aligned}$$

Here the first stage decision variables are x_{ij}, the number of aircraft of each type to assign to each route. There are 17 stage 1 variables and four constraints involving only stage 1 variables. The random variables in the problem are the customer demand d_j, and the stage 2 recourse variables are y_j and z_j, the number of bumped passengers and empty seats on each route respectively. There are 10 second stage variables and 5 random variables (which correspond to the right-hand sides of the five second stage constraints).

The second stage problem is

$$\begin{aligned}
\text{Minimize } & \sum_{j=1}^{5} e_j y_j \\
\text{subject to: } & y_j - z_j = b_j, \quad \forall j \in J \\
& y_j, z_j \geq 0
\end{aligned}$$

where $b_j := d_j - \sum_{i=1}^{4} p_{ij}x_{ij}$. Note that b_j is just the total customer demand for route j minus the total number of seats available for route j. In order to assess each random variable's

impact on the objective, we write the second stage dual problem :

$$\text{Maximize } \sum_{j=1}^{5} \pi_j b_j$$

$$\text{subject to: } 0 \leq \pi_j \leq e_j, \quad \forall j \in J$$

Using the demand distributions from Linderoth et al. (2005), this stochastic program has a total of 646425 random scenarios.

This turns out to be an easy problem to solve as there are no interdependencies between any of the constraints, and we can calculate exactly each random variable's contribution to the overall variance of the objective function. In the optimal solution, $\pi_j = e_j$ if $b_j > 0$ and $\pi_j = 0$ otherwise. Thus, the optimal objective value is $\sum_{j=1}^{5} e_j b_j 1_{\{b_j > 0\}}$. Since the five terms in the sum are all independent, the variance of the sum is equal to the sum of the variance, meaning that random variable d_j contributes $V_j = e_j \text{Var}(b_j 1_{\{b_j > 0\}})$ to the overall variance of the optimal objective value. Using the random demand scenarios from Linderoth et al. (2005), we can calculate the actual variance contributions at the optimal stage 1 solution. From Table 5.1, we can see that the most important random variable is d_4 followed by d_1. Further, with independence, we expect that the importance heuristics in Section 5.4 should all give similar estimates for the $\hat{V}_j$ as the non-diagonal terms in the covariance matrix should tend to zero as the number of samples increases.

LandS

Our next example is an electrical investment planning problem that originally appeared in Louveaux and Smeers (1988). The first stage decision involves allocating capacity to four new technologies subject to capacity minimums and budget restrictions. Thus, there are four stage 1 variables and two constraints involving only the stage 1 variables. The second

Table 5.1: Actual contribution to total variance (at optimal solution) - gbd

Random Variable	V_j
d_1	$134,670$
d_2	$5,035$
d_3	$11,440$
d_4	$292,319$
d_5	32

stage decision is regarding the production of three different modes of electricity for each technology to meet customer demand subject to the capacities from the first stage. There are 12 second stage decision variables, four second stage balance constraints (one for each technology), and three second stage demand constraints (one for each mode of electricity). In this problem the customer demand for each mode of electricity is random giving a total of three random variables. We use again use the demand distributions from Linderoth et al. (2005) where the random variables have identical independent distributions with 100 equally likely realizations yield a total of one million random scenarios. Notation for the problem is

as follows:

$$
\begin{aligned}
\text{Sets: } & I = \{1,2,3,4\} := \text{ Set of technologies} \\
& J = \{1,2,3\} := \text{ Set of electricity modes} \\
\text{Variables: } & x_i := \text{ Capacity of technology } i \\
& y_{ij} := \text{ Production of electricity mode } j \text{ for technology } i \\
\text{Parameters: } & K := \text{ Minimum required total capacity} \\
& B := \text{ Maximum allowable budget} \\
& b_i := \text{ Budget amount per unit capacity of technology } i \\
& f_i := \text{ Cost per unit capacity of technology } i \\
& c_{ij} := \text{ Cost per unit produced of electricity mode } j \text{ for technology } i \\
& d_j := \text{ Customer demand for electricity mode } j \text{ (random)}
\end{aligned}
$$

The problem can then be written:

$$
\begin{aligned}
\text{Minimize } & \sum_{i=1}^{4} f_i x_i + \sum_{i=1}^{4}\sum_{j=1}^{3} c_{ij} y_{ij} \\
\text{subject to: } & \sum_{i=1}^{4} x_i \geq K \\
& \sum_{i=1}^{4} b_i x_i \leq B \\
& -x_i + \sum_{j=1}^{3} y_{ij} \leq 0, \quad \forall i \in I \\
& \sum_{i=1}^{4} y_{ij} \geq d_j, \quad \forall j \in J \\
& x_i, y_{ij} \geq 0 \quad \forall i \in I, \forall j \in J
\end{aligned}
$$

The second stage primal problem is

$$\begin{aligned}
\text{Minimize } & \sum_{i=1}^{4}\sum_{j=1}^{3} c_{ij}y_{ij} \\
\text{subject to: } & \sum_{j=1}^{3} y_{ij} \leq x_i, \quad \forall i \in I \\
& \sum_{i=1}^{4} y_{ij} \geq d_j, \quad \forall j \in J \\
& y_{ij} \geq 0, \quad \forall i \in I, \forall j. \in J
\end{aligned}$$

If we associate dual multipliers σ_i to the capacity constraints and π_j to the demand constraints, the second stage dual problem is

$$\begin{aligned}
\text{Maximize } & \sum_{j=1}^{3} d_j\pi_j - \sum_{i=1}^{4} x_i\sigma_i \\
\text{subject to: } & \pi_j - \sigma_i \leq c_{ij} \quad \forall i \in I, \forall j \in J \\
& \sigma_i, \pi_j \geq 0 \quad \forall i \in I, \forall j \in J.
\end{aligned}$$

While the random variables $\{d_j, j = 1, 2, 3\}$ are each represented in the objective function, there are also terms with the dual multipliers σ_i which contain no random variables but are correlated with the π_j multipliers through the dual constraints.

In the original version of the problem, the cost coefficients c_{ij} in the primal objective function take the form $c_{i2} = 6c_{i3}$ and $c_{i1} = 10c_{i3}$. Since the demand random variables are identically distributed, the magnitude of the dual objective terms $d_j\pi_j$ are completely driven by the π_j, which in turn depend greatly on the level of the c_{ij} in the dual constraints. As a result, we find that nearly always the most important random variable is d_1, followed by d_2 and then d_3. The difference in the importance levels between the random variables is so

drastic that the importance heuristic is almost completely irrelevant. Thus to alleviate the effects of the structure of the c_{ij}, we will instead consider a second form of the problem where we shuffle the 12 c_{ij} coefficients. The rest of the problem's parameters remain the same.

apl1p

Our third problem is a model for electric power capacity expansion first described by Infanger (1992). The problem consists of two electrical generators that can operate at three levels. The first stage problem is to determine capacity for the two generators. The second stage decision is to determine how much to produce at each of the three operation levels on the two generators subject to (1) the availability of the two generators and (2) customer demand for each of the three operation levels. Thus, there are nine second stage variables and five second stage constraints. The availabilities of the generators and the customer demands for each level are all random giving a total of five random variables. Notation for the problem

is as follows:

$$
\begin{aligned}
\text{Sets: } & I = \{1, 2\} := \text{ Set of generators} \\
& J = \{1, 2, 3\} := \text{ Set of operation levels} \\
\text{Variables: } & x_i := \text{ Capacity of generator } i \\
& y_{ij} := \text{ Production at operation level } j \text{ on generator i} \\
& s_j := \text{ Unserved demand at operation level } j \\
\text{Parameters: } & c_i := \text{ Investment cost per unit capacity for generator } i \\
& f_{ij} := \text{ Operating cost per unit at operation level } j \text{ on generator } i \\
& u_j := \text{ Cost per unit of unserved demand at operation level } j \\
& K_i := \text{ Minimum required capacity for generator } i \\
& a_i := \text{ Availability of generator } i \text{ (random)} \\
& d_j := \text{ Customer demand for operation level } j \text{ (random)}
\end{aligned}
\tag{5.29}
$$

The problem can then be written:

$$
\begin{aligned}
\text{Minimize } & \sum_{i=1}^{4} c_i x_i + \sum_{i=1}^{2} \sum_{j=1}^{3} f_{ij} y_{ij} + \sum_{j=1}^{3} u_j s_j \\
\text{subject to: } & x_i \geq K_i, \quad \forall i \in I \\
& -a_i x_i + \sum_{j=1}^{3} y_{ij} \leq 0, \quad \forall i \in I \\
& \sum_{i=1}^{2} y_{ij} + s_j \geq d_j, \quad \forall j \in J \\
& x_i, y_{ij}, s_j \geq 0 \quad \forall i \in I, \forall j \in J.
\end{aligned}
$$

The second stage primal problem is

$$\begin{aligned}
\text{Minimize } & \sum_{i=1}^{2}\sum_{j=1}^{3} f_{ij}y_{ij} + \sum_{j=1}^{3} u_j s_j \\
\text{subject to: } & \sum_{j=1}^{3} y_{ij} \leq a_i x_i, \quad \forall i \in I \\
& \sum_{i=1}^{2} y_{ij} + s_j \geq d_j, \quad \forall j \in J \\
& y_{ij}, s_j \geq 0 \quad \forall i \in I, \forall j \in J.
\end{aligned}$$

If we again associate dual multipliers σ_i to the capacity constraints and π_j to the demand constraints, the second stage dual problem is

$$\begin{aligned}
\text{Maximize } & \sum_{j=1}^{3} d_j \pi_j - \sum_{i=1}^{2} a_i x_i \sigma_i \\
\text{subject to: } & \pi_j - \sigma_i \leq f_{ij} \quad \forall i \in I, \forall j \in J \\
& \pi_j \leq u_j \quad \forall j \in J \\
& \sigma_i, \pi_j \geq 0 \quad \forall i \in I, \forall j \in J.
\end{aligned}$$

Again, the terms in the dual objective function are correlated due to the structure of the dual constraints. Using the probability distributions for the availabilities and demands from Infanger (1992) the problem has 1280 random scenarios. Given the small number of scenarios, we can write out and solve the deterministic equivalent of this stochastic program and also evaluate the second stage objective function at every scenario given the optimal stage 1 solution. This enables us to perform some of the sensitivity analysis calculations from Section 5.3 that are too cumbersome for large problems. Table 5.2 contains the calculations of the correlation ratio (Equation 5.16) and the Pearson correlation coefficients, or PEAR,

Table 5.2: Estimating important variables - apl1p

Random Variable	Corr. Ratio	PEAR
d_1	0.0340	0.1844
d_2	0.0255	0.1597
d_3	0.0132	0.1146
a_1	0.5154	0.7153
a_2	0.3976	0.6244

(Equation 5.15) from the total enumeration of the stochastic program at the optimal stage 1 solution. We can see that the availability random variables (a_1 and a_2) are the most important given the first stage optimal solution.

20term

Our final problem is *20term* – a vehicle allocation problem for a motor freight carrier that appeared in Bailey et al. (1999). The company has a central depot and 20 outlying terminals and has a fleet consisting of three types of tractors and trailers it must allocate to the 21 stations. The first stage problem, which contains 63 decision variables and three constraints, is to decide on the fleet configuration at the beginning of the day. The second stage problem is a vehicle routing problem where random point-to-point demands for shipments must be satisfied (incurring penalties for unsatisfied demands). Additionally, the fleet configuration at the end of the day must match the starting configuration. The second stage problem has 764 decision variables and 124 constraints. Forty of these constraints correspond to satisfying the point-to-point demands and have random right-hand sides. Each random demand has a bimodal probability distribution which gives the problem $2^{40} = 10^{12}$ scenarios. We modified the probability distributions from the original problem by making the mean μ_j of the right-hand side of the j^{th} constraint a uniformly distributed integer between 20 and 60 and the

deviation σ_j a uniformly distributed integer between 10 and 20 (yielding the equally likely values $\mu_j - \sigma_j$ and $\mu_j + \sigma_j$).

5.7.2 Numerical Results of the Algorithm

We performed the following combinations of runs to test our algorithms:

- 4 test problems: gbd, LandS, apl1p, 20term
- 5 algorithms: ES-PMC, ES-PLHS, ES-MC, ES-LHS, ES-QMC
- 2 stopping criteria: 1RP, I2RP
- 3 importance heuristics: Heuristic 1 (Main), Heuristic 2 (Total), and Heuristic 3 (PCA). Note that these heuristics are only needed for the ES-PMC and ES-PLHS algorithms.

We used three measures to assess the performance of the algorithms:

1. The optimal value at the end of the algorithm
2. The number of iterations of the algorithm until convergence
3. The run time of the algorithm

Note that in the algorithms with padding, since we need to estimate the important subset, each iteration uses twice as many samples as an iteration of an algorithm without padding. For each run combination, we ran 10 independent replications in order to calculate confidence intervals for our performance measures. Parameters used in the algorithm were $\epsilon = 0.002$, $n_0 = 8$, and $\kappa = 2$. We report the mean and the standard deviation for each performance measure. Results can be found in the tables in Appendix A.

From our results, we see that the pure Monte Carlo algorithm ES-MC requires more iterations, takes longer to run, and has a higher variance on the optimal value than the

non-i.i.d. sampling algorithms. We also see that for the larger problem 20term, the ES-QMC algorithm has a significantly longer run time than the other algorithms, highlighting the need for some form of padding in a QMC sampling algorithm. Further, we see that ES-PLHS performs better than ES-PMC in nearly all areas. The run times are faster and the confidence intervals on the optimal values are tighter. The one exception here is in the problem gbd (Table A.1) where running ES-PMC for three times the number of iterations as ES-PLHS improves the accuracy of the optimal values.

As for the stopping criteria, the 1RP has much faster run times than the I2RP. This again is not surprising as the I2RP involves solving an extra stochastic program at each iteration as the mean and standard deviation of the optimality gap are estimated using two separate point sets. The confidence intervals on the optimal values are slightly tighter for the I2RP but the mean optimal values are almost identical. Given the drastic difference in run times, it is advisable to use the 1RP stopping criterion.

The three importance heuristics are indistinguishable on the gbd and LandS problems. The results vary more on the other two problems, but are not yet significantly different enough to make a clear choice between the heuristics.

5.8 Conclusions

In this chapter, we developed an adaptive sampling algorithm that uses padded sampling to solve two-stage stochastic linear programs. In addition, using the central limit theorem for PLHS that we proved in Chapter 4, we showed that stopping criteria for the algorithm where padded Latin Hypercube sampling is used to construct a confidence interval around the optimality gap are asymptotically valid. Similarly, central limit theorem results for LHS and PMC sampling enable us to prove corresponding results for stopping criteria that use those sampling methods. While our numerical results show significant advantages of PLHS over

pure Monte Carlo, pure QMC, and PMC sampling for our specific test problems, the results of ES-LHS and ES-PLHS are not statistically different with 10 replications. However, the run times of ES-PLHS begin to look consistently faster for ES-PLHS on the 20term problem. We expect as solve larger stochastic programs that this difference in run time will be more pronounced. This is an area we hope to explore further.

Chapter 6

Conclusion

As we have noted, solving stochastic programs using Monte Carlo sampling has been well studied. In this dissertation, we have focused on different Quasi-Monte Carlo strategies designed to reduce the number of samples required to solve a stochastic program. One of these strategies was Latin Hypercube sampling. In Chapter 3, we showed that under certain assumptions, the probability that the sample mean of a function under Latin Hypercube differs from its true mean by some value δ is no larger than the corresponding deviation probability under Monte Carlo sampling. These are the only known large deviations results specific to Latin Hypercube sampling. When these results are applied to stochastic programming problems, they yield that on average the number of samples needed to obtain a particular solution accuracy is smaller for LHS than for Monte Carlo sampling.

Another strategy we looked at was padded sampling. Since pure (R)QMC sampling can lose its effectiveness in high dimensions, we chose instead to only perform (R)QMC on an important subset of the random variables and to pad the rest of the sample with some other sampling method like Monte Carlo or LHS. When padding is done with LHS (called PLHS), we have shown in Chapter 4 that an asymptotic normal distribution holds. Again, this is the first known central limit theorem proven for PLHS. Also, analogous to the relationship

between pure Monte Carlo and pure LHS, the asymptotic variance under PLHS is no worse than that under PMC (padding done with Monte Carlo). Now that we have a central limit theorem for general PLHS, we hope to derive convergence results for the optimal values and optimal solutions of sampled stochastic programs.

In Chapter 5, we proposed a sampling algorithm ES-PAD to solve two stage stochastic linear programs using padded sampling. While other papers have looked at solving stochastic programs using QMC, this is the first proposed algorithm to do so in an adaptive way – as the subset of important variables in the padded sampling scheme is redetermined each time a new first stage solution is calculated.

One area from this thesis we would like to explore more is choosing the subset of important variables for the padded sampling scheme. There are a number of sensitivity analysis techniques available, however, we are limited in that our function evaluations involve solving stochastic programs, and thus are expensive. We developed some heuristics for quickly determining the importance of a random variable, and while they seem to perform well, they perform very similarly on our small- and medium-sized problems. It is likely when solving very large stochastic programs that these heuristics will need to be further refined.

Another area we did not explore much is focusing not only on reducing the number of required samples to solve the stochastic program, but also on reducing the actual time spent solving the program, i.e., the wall clock time. While these often go hand in hand, there are other ways to reduce the wall clock time. Should we find that the accuracy in estimating the important subset is critical when solving very large problems, we may face the decision of using more computing time to estimate the importance of each random variable versus using extra samples to solve the problem with a rough importance heuristic. It is worth noting though that using our current software (`sutil` and `ATR`), the time spent estimating the importance of each random variable using our heuristics is negligible compared to the

time spent solving the stochastic program.

Additionally, when we begin solving very large stochastic programs, it may be necessary to modify our ES-PAD algorithm to run on a computational grid. Such algorithms employ a Master-Worker (MW) paradigm where many machines work on different pieces of the algorithm in parallel and report the results to one Master machine. Linderoth and Wright (2001) have used Algorithm ATR to solve stochastic programs using Monte Carlo and Latin Hypercube sampling on a computational grid and reported significant savings in wall clock time. It remains to be seen whether our padded algorithms can also be effectively parallelized.

Bibliography

T. Andres. Sampling methods and sensitivity analysis for large parameter sets. *J. Statist. Comput.*, 57:77–110, 1997.

T. Andres and W. Hajas. Using iterated fractional factorial design to screen paramters in sensitivity analysis of a probabilistic risk assessment model. In K editor, *Proceedings of the Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications*, Karlsruhe, Germany, April 1993.

T. G. Bailey, P. Jensen, and D. Morton. Response surface analysis of two-stage stochastic linear programming with recourse. *Naval Research Logistics*, 46:753–778, 1999.

R. Bartle. *The Elements of Real Analysis*. Wiley, New York, 2nd. edition, 1987.

G. Bayraksan and D. Morton. Assessing solution quality in stochastic programs. *Mathematical Programming*, 108:495–514, 2006.

G. Bayraksan and D. Morton. A sequential sampling procedure for stochastic programming. Under Revision, 2007.

G. Bayraksan, D. Morton, and A. Partani. Simulation-based optimality tests for stochastic programs. In G. Dantzig and G. Infanger, editors, *Planning Under Uncertainty: Stochastic Programming*. Kluwer Series on Advances in Mathematical Programming, 2007.

E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B*, 17:173–184, 1955.

B. Bettonvil. *Detection of Important Factors by Sequential Bifurcation*. Tilburg University Press, Tilburg, 1990.

B. Bettonvil and J. Kleijnen. Searching for important factors in simulation models with many factors: Sequential bifurcation. *Eur. J. Oper. Res.*, 96:180–194, 1997.

J. A. Bucklew. *Introduction to Rare Event Simulation*. Springer-Verlag, New York, 2004.

R. E. Caflisch, W. J. Morokoff, and A. B. Owen. Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *Journal of Computational Finance*, 1:27–46, 1997.

A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 5: 73–79, 1959.

K. L. Chung. *A Course in Probability Theory*. Academic Press, New York, NY, 2nd. edition, 1974.

R. Cranley and T. Patterson. Randomization for number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914, 1976.

R. Cukier, C. Fortuin, K. Shuler, A. Petschek, and J. Schaibly. A study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i. theory. *J. Chem. Phys.*, 59:3873–3878, 1973.

J. Czyzyk, J. Linderoth, and J. Shen. SUTIL: A utility library for handling stochastic programs, 2005. User's Manual. Software available at `http://coral.ie.lehigh.edu/sutil`.

L. Dai, C. H. Chen, and J. R. Birge. Convergence properties of two-stage stochastic programming. *J. Optim. Theory Appl.*, 106(3):489–509, 2000.

G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1963.

G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

G. B. Dantzig and P. W. Glynn. Parallel processors for planning under uncertainty. *Annals of Operations Research*, 22:1–21, 1990.

A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer-Verlag, New York, NY, 2nd. edition, 1998.

F. den Hollander. *Large Deviations*. Number 14 in Fields Institute Monographs. American Mathematical Society, Providence, RI, 2000.

S. S. Drew and T. Homem-de-Mello. Some large deviations results for latin hypercube sampling. Manuscript, Department of Industrial Engineering and Management Sciences, Northwestern University, 2005.

J. Dupačová and R. J.-B. Wets. Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. *The Annals of Statistics*, 16:1517–1549, 1988.

M. Emsermann and B. Simon. Improving simulation efficiency with quasi control variates. Manuscript, 2000.

Y. Ermoliev. Stochastic quasi-gradient methods and their application to systems optimization. *Stochastics*, 4:1–37, 1983.

B. L. Fox. *Strategies for Quasi-Monte Carlo*. Kluwer Academic Publishers, Norwell, MA, 2000.

I. Friedel and A. Keller. Fast generation of randomized low-discrepancy point sets. In *Monte Carlo and quasi-Monte Carlo methods, 2000 (Hong Kong)*, pages 257–273. Springer, Berlin, 2002. Software available at `http://www.multires.caltech.edu/software/libseq/`.

G. Gürkan, A. Y. Özge, and S. M. Robinson. Sample-path solutions of stochastic variational inequalities. *Mathematical Programming*, 84:313–334, 1999.

J. Hammersley. Monte carlo methods for solving multivariable problems. *Ann. New York Acad. Sci.*, 86:844–874, 1960.

G. H. Hardy. On double fourier series, and especially those which represent the double zeta-function with real and incommensurable parameters. *Quarterly Journal of Mathematics*, 37:53–79, 1905.

J. L. Higle. Variance reduction and objective function evaluation in stochastic linear programs. *INFORMS Journal on Computing*, 10(2):236–247, 1998.

J. L. Higle and S. Sen. Stochastic decomposition: An algorithm for two stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

J. B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms*, volume II. Springer-Verlag, Berlin, Germany, 1993.

T. Homem-de Mello. On rates of convergence for stochastic optimization problems under non-i.i.d. sampling. Manuscript, 2006.

T. Homma and A. Saltelli. Global sensitivity analysis of nonlinear models: Importance measures and sobol' sensitivity indices. Technical Report EUR 16052/EN, European Commission: Joint Research Centre, 1994.

T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering and System Safety*, 52:1–17, 1996.

S. Hora and R. Iman. A comparison of maximum/bounding anf bayesian/monte carlo for fault tree uncertainty analysis. Technical Report Technical Report: SAND85-2839, Sandia National Laboratories, Albuquerque, NM, 1986.

N. Hoshino and A. Takemura. On reduction of finite sample variance by extended latin hypercube sampling. *Bernoulli*, 6(6):1035–1050, 2000.

R. Iman and S. Hora. A robust measure of uncertainty importance for use in fault tree system analysis. *Risk Analysis*, 10:401–406, 1999.

G. Infanger. Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs. *Annals of Operations Research*, 39:69–95, 1992.

G. Infanger. *Planning under Uncertainty: Solving Large Scale Stochastic Linear Programs*. Boyd & Fraser Publishing Company, Massachusetts, 1994.

X. Jin, M. C. Fu, and X. Xiong. Probabilistic error bounds for simulation quantile estimators. *Management Science*, 49(2):230–246, 2003.

J. Kalagnanam and U. Diwekar. An efficient sampling technique for off-line quality control. *Technometrics*, 39(3):308–319, 1997.

Y. M. Kaniovski, A. J. King, and R. J.-B. Wets. Probabilistic bounds (via large deviations) for the solutions of stochastic programming problems. *Ann. Oper. Res.*, 56:189–208, 1995.

A. J. King and R. T. Rockafellar. Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, 18:148–162, 1993.

A. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2001.

M. Koivu. Variance reduction in sample approximations of stochastic programs. *Mathematical Programming*, pages 463–485, 2005.

A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, NY, 3rd. edition, 2000.

P. L'Ecuyer and C. Lemieux. Recent advances in randomized quasi-Monte Carlo methods. In M. Dror, P. L'Ecuyer, and F. Szidarovszky, editors, *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, pages 419–474. Kluwer Academic Publishers, Boston, 2002.

J. Linderoth and S. Wright. Decomposition algorithms for stochastic programming on a computational grid. Manuscript, available on *Optimization Online* (`www.optimization-online.org`), 2001.

J. T. Linderoth, A. Shapiro, and S. J. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 2005. forthcoming.

R. Liu and A. B. Owen. Estimating mean dimensionality. Manuscript, Department of Statistics, Stanford University, 2003.

W. Loh. On latin hypercube sampling. *The Annals of Statistics*, 24(5):2058–2080, 1996.

F. Louveaux and Y. Smeers. Optimal investments for electricity generation: A stochastic model and a test problem. In Y. Ermoliev and R. J.-B. Wets, editors, *Numerical techniques for stochastic optimization problems*, pages 445–452. Springer-Verlag, Berlin, 1988.

W. K. Mak, D. P. Morton, and R. K. Wood. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47–56, 1999.

M. McKay. Evaluating prediction uncertainty. Technical Report Technical Report NUREG/CR-6311, U.S. Nuclear Regulatory Commission and Los Alamos National Laboratory, 1995.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

M. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33:161–174, 1991.

H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA, 1992.

G. Ökten. High-dimensional simulation. *Mathematics and Computers in Simulation*, 55: 215–222, 2001.

G. Ökten. A probabilistic result on the discrepancy of a hybrid-monte carlo sequence and applications. *Monte Carlo Methods and Applications 2*, 4:255–270, 1996.

G. Ökten, B. Tuffin, and V. Burago. A central limit theorem and improved error bounds for a hybrid-monte carlo sequence with applications in computational finance. *Journal of Complexity*, 22:435–458, 2006.

A. B. Owen. A central limit theorem for latin hypercube sampling. *J. Roy. Statist. Soc. Ser. B*, 54:541–551, 1992.

A. B. Owen. Monte Carlo, quasi-Monte Carlo, and randomized quasi-Monte Carlo. In *Monte Carlo and quasi-Monte Carlo methods 1998 (Claremont, CA)*, pages 86–97. Springer, Berlin, 2000.

A. B. Owen. Necessity of low effective dimension. Manuscript, Department of Statistics, Stanford University, 2002.

A. B. Owen. The dimension distribution and quadrature test functions. *Statist. Sinica*, 13 (1):1–17, 2003.

A. B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and quasi-Monte Carlo methods in scientific computing (Las Vegas, NV, 1994)*, volume 106 of *Lecture Notes in Statist.*, pages 299–317. Springer, New York, 1995.

A. B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM J. Numer. Anal.*, 34 (5):1884–1910, 1997a.

A. B. Owen. Scrambled net variance for integrals of smooth functions. *Ann. Statist.*, 25(4): 1541–1562, 1997b.

A. B. Owen. Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation*, 8:71–102, 1998.

T. Pennanen. Epi-convergent discretizations of multistage stochastic programs. *Mathematics of Operations Research*, 30:245–256, 2005.

T. Pennanen and M. Koivu. Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische Mathematik*, 100:141–163, 2005.

E. L. Plambeck, B. R. Fu, S. M. Robinson, and R. Suri. Sample-path optimization of convex stochastic performance functions. *Mathematical Programming, Series B*, 75:137–176, 1996.

H. Robbins and S. Monro. On a stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 1951a.

H. Robbins and S. Monro. On a stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951b.

S. M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21:513–528, 1996.

H. L. Royden. *Real Analysis*. Macmillan, New York, 3rd edition, 1988.

R. Y. Rubinstein and A. Shapiro. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. J. Wiley & Sons, Chichester, England, 1993.

A. Saltelli, S. Tarantola, and K. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41:39–56, 1999.

A. Saltelli, K. Chan, and E. Scott. *Sensitivity Analysis*. Wiley, New York, 2000.

A. Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, 30: 169–186, 1991.

A. Shapiro. Asymptotic behavior of optimal solutions in stochastic programming. *Mathematics of Operations Research*, 18:829–845, 1993.

A. Shapiro and T. Homem-de-Mello. On rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11:70–86, 2000.

A. Shapiro and T. Homem-de-Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81:301–325, 1998.

A. Shapiro and A. Nemirovski. On complexity of stochastic programming problems. Manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, 2004.

A. Shapiro, T. Homem-de-Mello, and J. C. Kim. Conditioning of convex piecewise linear stochastic programs. *Mathematical Programming*, 94:1–19, 2002.

I. Sobol. Sensitivity analysis for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1:407–414, 1993.

J. Spanier. Quasi-monte carlo methods for particle transport problems. In H. Niederreiter and P. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 121–148. Springer-Verlag, New York, 1995.

M. L. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29:143–151, 1987.

A. Tamhane and D. Dunlop. *Statistics and Data Analysis from Elementary to Intermediate*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

B. Tuffin. On the use of low-discrepancy sequences in monte carlo methods. Technical Report Technical Report No. 1060, I.R.I.S.A., Rennes, France, 1996.

Appendix A. Numerical Results for ES-PAD and ES-NoPAD

Table A.1: A comparison of sampling algorithms – gbd

	Stopping	*Importance*	*Optimal Value*		*Iterations*		*Run Time*	
Algorithm	*Criteria*	*Heuristic*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
PAD-MC	1RP	Main	1661	12	4.3	1.5	0:05:19	0:02:23
PAD-MC	I2RP	Main	1652	8	4.5	1.1	0:08:17	0:02:33
PAD-MC	1RP	Total	1661	12	4.3	1.5	0:05:18	0:02:22
PAD-MC	I2RP	Total	1652	8	4.5	1.1	0:08:15	0:02:34
PAD-MC	1RP	PCA	1661	12	4.3	1.5	0:05:16	0:02:22
PAD-MC	I2RP	PCA	1652	8	4.5	1.1	0:08:12	0:02:32
PAD-LHS	1RP	Main	1666	29	1.5	0.5	0:01:41	0:00:32
PAD-LHS	I2RP	Main	1666	29	1.5	0.5	0:02:27	0:00:46
PAD-LHS	1RP	Total	1666	29	1.5	0.5	0:01:41	0:00:32
PAD-LHS	I2RP	Total	1666	29	1.5	0.5	0:02:26	0:00:46
PAD-LHS	1RP	PCA	1666	29	1.5	0.5	0:01:41	0:00:32
PAD-LHS	I2RP	PCA	1666	29	1.5	0.5	0:02:26	0:00:46
MC	1RP	Main	1666	39	5.1	0.7	0:05:57	0:01:12
MC	I2RP	Main	1686	36	5.1	0.7	0:08:57	0:01:41
LHS	1RP	Main	1663	24	2.0	0.0	0:02:01	0:00:06
LHS	I2RP	Main	1663	24	2.0	0.0	0:03:02	0:00:07
QMC	1RP	Main	1653	35	2.1	0.7	0:02:11	0:00:43
QMC	I2RP	Main	1653	35	2.1	0.7	0:03:16	0:01:09
PAD-MC	All	All	1656	11	4.4	1.3	0:06:46	0:02:47
PAD-LHS	All	All	1666	28	1.5	0.5	0:02:04	0:00:45
MC	All	All	1676	38	5.1	0.7	0:07:27	0:02:06
LHS	All	All	1663	24	2.0	0.0	0:02:32	0:00:32
QMC	All	All	1653	34	2.1	0.7	0:02:44	0:01:05
All	1RP	All	1662	26	3.0	1.7	0:03:27	0:02:18
All	I2RP	All	1662	26	3.0	1.7	0:05:15	0:03:17
All	All	Main	1661	22	3.0	1.8	0:04:26	0:03:09
All	All	Total	1661	22	3.0	1.8	0:04:25	0:03:09
All	All	PCA	1661	22	3.0	1.8	0:04:24	0:03:08
All	All	All	1662	26	3.0	1.7	0:04:21	0:02:58
True Optimal Value = 1656								

Table A.2: A comparison of sampling algorithms – LandS

	Stopping	*Importance*	*Optimal Value*		*Iterations*		*Run Time*	
Algorithm	*Criteria*	*Heuristic*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
PAD-MC	1RP	Main	128.32	0.80	1.9	0.7	0:01:32	0:00:31
PAD-MC	I2RP	Main	128.26	0.78	2.0	0.7	0:02:22	0:00:43
PAD-MC	1RP	Total	128.32	0.80	1.9	0.7	0:01:31	0:00:31
PAD-MC	I2RP	Total	128.26	0.78	2.0	0.7	0:02:21	0:00:43
PAD-MC	1RP	PCA	128.32	0.80	1.9	0.7	0:01:32	0:00:31
PAD-MC	I2RP	PCA	128.20	0.81	2.1	0.7	0:02:29	0:00:49
PAD-LHS	1RP	Main	128.26	0.13	1.7	0.5	0:01:47	0:00:23
PAD-LHS	I2RP	Main	128.28	0.10	1.9	0.3	0:02:50	0:00:23
PAD-LHS	1RP	Total	128.26	0.13	1.7	0.5	0:01:47	0:00:21
PAD-LHS	I2RP	Total	128.28	0.10	1.9	0.3	0:02:50	0:00:23
PAD-LHS	1RP	PCA	128.26	0.13	1.7	0.5	0:01:47	0:00:23
PAD-LHS	I2RP	PCA	128.28	0.10	1.9	0.3	0:02:50	0:00:23
MC	1RP	Main	128.27	1.69	4.6	1.2	0:03:42	0:01:10
MC	I2RP	Main	128.66	1.27	4.7	1.3	0:05:48	0:01:55
LHS	1RP	Main	128.09	0.44	1.6	0.5	0:01:12	0:00:20
LHS	I2RP	Main	128.03	0.36	1.7	0.5	0:01:54	0:00:29
QMC	1RP	Main	128.19	0.15	2.3	0.7	0:01:43	0:00:33
QMC	I2RP	Main	128.19	0.15	2.3	0.7	0:02:35	0:00:52
PAD-MC	All	All	128.28	0.76	2.0	0.7	0:01:58	0:00:45
PAD-LHS	All	All	128.27	0.11	1.8	0.4	0:02:19	0:00:38
MC	All	All	128.47	1.47	4.7	1.2	0:04:45	0:01:53
LHS	All	All	128.06	0.39	1.7	0.5	0:01:33	0:00:32
QMC	All	All	128.19	0.14	2.3	0.7	0:02:09	0:00:50
All	1RP	All	128.26	0.72	2.1	1.1	0:01:50	0:00:53
All	I2RP	All	128.27	0.63	2.3	1.1	0:02:53	0:01:21
All	All	Main	128.28	0.54	1.9	0.6	0:02:08	0:00:43
All	All	Total	128.28	0.54	1.9	0.6	0:02:07	0:00:43
All	All	PCA	128.27	0.55	1.9	0.6	0:02:10	0:00:45
All	All	All	128.26	0.67	2.2	1.1	0:02:22	0:01:15
True Optimal Value = 128.20								

Table A.3: A comparison of sampling algorithms – apl1p

	Stopping	*Importance*	*Optimal Value*		*Iterations*		*Run Time*	
Algorithm	*Criteria*	*Heuristic*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
PAD-MC	1RP	Main	24626	194	4.5	1.7	0:08:21	0:02:51
PAD-MC	I2RP	Main	24579	136	5.4	1.3	0:14:25	0:03:09
PAD-MC	1RP	Total	24689	115	3.4	0.5	0:06:34	0:01:03
PAD-MC	I2RP	Total	24678	125	3.3	0.5	0:09:03	0:01:18
PAD-MC	1RP	PCA	24581	142	3.6	0.7	0:06:45	0:00:58
PAD-MC	I2RP	PCA	24581	142	3.6	0.7	0:09:50	0:01:33
PAD-LHS	1RP	Main	24664	138	2.7	1.3	0:05:30	0:02:01
PAD-LHS	I2RP	Main	24667	141	2.8	1.2	0:08:23	0:02:52
PAD-LHS	1RP	Total	24664	139	2.8	1.3	0:05:36	0:02:05
PAD-LHS	I2RP	Total	24667	141	2.8	1.3	0:08:13	0:03:16
PAD-LHS	1RP	PCA	24664	136	2.6	1.1	0:05:13	0:01:43
PAD-LHS	I2RP	PCA	24668	139	2.6	1.1	0:07:41	0:02:39
MC	1RP	Main	24720	232	4.9	1.4	0:08:57	0:02:18
MC	I2RP	Main	24712	247	5.6	1.2	0:15:05	0:03:10
LHS	1RP	Main	24593	182	2.8	1.2	0:05:10	0:01:52
LHS	I2RP	Main	24639	76	3.2	0.9	0:08:19	0:01:59
QMC	1RP	Main	24659	206	3.2	0.8	0:05:47	0:01:12
QMC	I2RP	Main	24688	170	3.4	0.7	0:08:56	0:01:35
PAD-MC	All	All	24622	146	4.0	1.2	0:09:10	0:03:16
PAD-LHS	All	All	24666	133	2.7	1.2	0:06:46	0:02:45
MC	All	All	24716	233	5.3	1.3	0:12:01	0:04:09
LHS	All	All	24616	138	3.0	1.1	0:06:44	0:02:29
QMC	All	All	24674	184	3.3	0.7	0:07:21	0:02:07
All	1RP	All	24651	167	3.4	1.4	0:06:26	0:02:13
All	I2RP	All	24653	152	3.6	1.4	0:09:59	0:03:33
All	All	Main	24634	153	3.9	1.8	0:09:10	0:04:13
All	All	Total	24674	126	3.1	1.0	0:07:22	0:02:27
All	All	PCA	24624	141	3.1	1.0	0:07:22	0:02:26
All	All	All	24652	159	3.5	1.4	0:08:13	0:03:27
True Optimal Value = 24642								

Table A.4: A comparison of sampling algorithms – 20term

Algorithm	*Stopping Criteria*	*Importance Heuristic*	*Optimal Value*		*Iterations*		*Run Time*	
			Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*
PAD-MC	1RP	Main	529287	2411	2.2	0.9	0:11:34	0:05:16
PAD-MC	I2RP	Main	529274	2538	2.3	0.7	0:17:08	0:05:21
PAD-MC	1RP	Total	529232	2242	2.0	0.7	0:10:17	0:03:27
PAD-MC	I2RP	Total	529232	2242	2.0	0.7	0:14:45	0:05:06
PAD-MC	1RP	PCA	530116	3829	2.7	0.8	0:14:09	0:04:49
PAD-MC	I2RP	PCA	530470	3120	2.9	0.9	0:22:52	0:08:10
PAD-LHS	1RP	Main	531410	2370	1.9	0.6	0:09:10	0:02:43
PAD-LHS	I2RP	Main	531601	2405	2.0	0.8	0:14:32	0:07:22
PAD-LHS	1RP	Total	530593	1603	1.7	0.7	0:08:20	0:03:11
PAD-LHS	I2RP	Total	530489	1532	1.9	0.7	0:13:29	0:05:24
PAD-LHS	1RP	PCA	531610	1265	1.9	0.9	0:09:30	0:04:12
PAD-LHS	I2RP	PCA	531404	1211	2.3	0.8	0:16:37	0:06:08
MC	1RP	Main	533251	5012	2.4	1.2	0:13:00	0:07:36
MC	I2RP	Main	533288	4325	2.6	0.7	0:19:31	0:06:38
LHS	1RP	Main	531052	1601	2.0	1.1	0:10:23	0:05:44
LHS	I2RP	Main	531214	1502	2.1	1.1	0:15:45	0:08:49
QMC	1RP	Main	529489	1537	5.1	1.0	0:33:57	0:12:01
QMC	I2RP	Main	529799	1437	5.0	0.8	0:46:34	0:12:12
PAD-MC	All	All	529602	2716	2.4	0.8	0:15:08	0:06:44
PAD-LHS	All	All	531185	1782	2.0	0.7	0:11:56	0:05:48
MC	All	All	533270	4556	2.5	0.9	0:16:16	0:07:42
LHS	All	All	531133	1513	2.1	1.1	0:13:04	0:07:44
QMC	All	All	529644	1457	5.1	0.9	0:40:16	0:13:27
All	1RP	All	530671	2859	2.4	1.3	0:13:22	0:09:30
All	I2RP	All	530752	2635	2.6	1.2	0:20:08	0:12:09
All	All	Main	530393	2595	2.1	0.7	0:13:06	0:06:03
All	All	Total	529886	1974	1.9	0.7	0:11:43	0:04:57
All	All	PCA	530900	2595	2.5	0.9	0:15:47	0:07:35
All	All	All	530712	2742	2.5	1.2	0:16:45	0:11:24
True Optimal Value = 531000±1000								