

ALGORITHMIC DEVELOPMENTS IN MONTE CARLO  
SAMPLING-BASED METHODS FOR STOCHASTIC  
PROGRAMMING

by

Péguy Pierre-Louis

---

A Dissertation Submitted to the Faculty of the  
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING  
In Partial Fulfillment of the Requirements  
For the Degree of  
DOCTOR OF PHILOSOPHY  
In the Graduate College  
THE UNIVERSITY OF ARIZONA

2012

THE UNIVERSITY OF ARIZONA  
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Péguy Pierre-Louis entitled Algorithmic Developments in Monte Carlo Sampling-Based Methods for Stochastic Programming and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

\_\_\_\_\_  
Güzin Bayraksan

Date: 18 April 2012

\_\_\_\_\_  
Larry Head

Date: 18 April 2012

\_\_\_\_\_  
Wei Lin

Date: 18 April 2012

\_\_\_\_\_  
Young-Jun Son

Date: 18 April 2012

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

\_\_\_\_\_  
Dissertation Director: Güzin Bayraksan

Date: 18 April 2012

### STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Péguy Pierre-Louis

## ACKNOWLEDGEMENTS

This work would not have been possible without the all around support of my advisor, Dr. Güzin Bayraksan. I am extremely grateful for Dr. Bayraksan's guidance. I have learned so much from the faculty in the SIE department, I am grateful for their support as well. I would also like to acknowledge the staff at SIE who have made the PhD journey a lot smoother. Lastly, I am very thankful for all the friends and colleagues I have made in the SIE department, the University of Arizona and the Tucson community in general.

## DEDICATION

*I dedicate this dissertation to my family: mom, dad, my brothers and sisters (Guerlie, Emyonne, Michel, Myriam and Alain). I am for ever thankful for their love and support.*

## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	8
LIST OF TABLES . . . . .	9
ABSTRACT . . . . .	10
CHAPTER 1 INTRODUCTION . . . . .	11
1.1 Motivation . . . . .	14
1.2 Contributions . . . . .	16
1.3 Definitions and Assumptions . . . . .	17
1.4 Dissertation Organization . . . . .	19
CHAPTER 2 FIXED-WIDTH SEQUENTIAL SAMPLING METHOD (FWSSM)	20
2.1 Literature Review . . . . .	21
2.2 Problem Class . . . . .	24
2.3 Confidence Interval Estimation of Optimality Gap . . . . .	25
2.4 Limit Theorems for Random Variables with Random Indexes . . . . .	28
2.5 Fully Sequential Procedure . . . . .	30
2.5.1 Description . . . . .	30
2.5.2 Asymptotic Validity, Finite Stopping, and Finite Sample Sizes . . . . .	33
2.5.3 Rate of increase of $M(\varepsilon)$ and $N(\varepsilon)$ . . . . .	37
2.5.4 Parameter Values of Linear Schedules . . . . .	40
2.6 Sequential Procedure with Stochastic Schedules . . . . .	42
2.6.1 Description . . . . .	42
2.6.2 Asymptotic Validity, Finite Stopping, and Finite Sample Sizes . . . . .	46
2.7 Summary and Concluding Remarks . . . . .	50
CHAPTER 3 FWSSM: COMPUTATIONAL RESULTS . . . . .	51
3.1 Test Problems: APL1P and 10D . . . . .	52
3.2 Effect of Resampling . . . . .	53
3.3 Results on APL1P and 10D . . . . .	57
3.4 Effect of $\varepsilon$ . . . . .	62
3.5 Selection of $\varepsilon$ and Results on 20TERM . . . . .	63
3.6 Summary and Concluding Remarks . . . . .	66

TABLE OF CONTENTS – *Continued*

CHAPTER 4	SAMPLING-BASED SEQUENTIAL APPROXIMATION METHOD . . .	69
4.1	Literature Review . . . . .	70
4.2	Problem Class . . . . .	74
4.3	Review of Sequential Approximation Method . . . . .	76
4.3.1	Bounding the Objective Function . . . . .	76
4.3.2	Generating Solutions . . . . .	77
4.3.3	Refining Partitions . . . . .	77
4.3.4	SAM and Convergence Properties . . . . .	79
4.4	Sampling-Based Sequential Approximation Method (SSAM) . . . . .	80
4.4.1	Stratified Monte Carlo Sampling . . . . .	81
4.4.2	Control Variates Scheme . . . . .	82
4.4.3	Rules to Increase the Sample Sizes and to Stop . . . . .	84
4.4.4	Partition Refinement . . . . .	85
4.4.5	Algorithm Statement . . . . .	86
4.5	Asymptotic Validity . . . . .	88
4.6	Summary and Concluding Remarks . . . . .	92
CHAPTER 5	SSAM: COMPUTATIONAL RESULTS . . . . .	93
5.1	Test Problems . . . . .	93
5.2	Parameter Settings . . . . .	95
5.3	Variance Reduction . . . . .	95
5.4	Empirical Results on the Asset Allocation Model . . . . .	97
5.5	Performance of SSAM as $d_\xi$ increases . . . . .	98
5.6	Summary and Concluding Remarks . . . . .	100
CHAPTER 6	CONCLUSIONS . . . . .	103
6.1	Summary of Contributions . . . . .	103
6.2	Future Research . . . . .	105
REFERENCES	. . . . .	107

## LIST OF FIGURES

2.1	Values of $c_0^*$ vs $\beta$ for FSP with (a) A1RP and (b) A2RP. . . . .	42
3.1	Probability of obtaining $\varepsilon$ -optimal solutions for (a) APL1P and (b) 10D. . .	52
3.2	Effect of resampling. . . . .	56
3.3	Coverage probabilities for (a) APL1P and (b) 10D. . . . .	58
3.4	Coverage probability as $\varepsilon$ shrinks. . . . .	64
5.1	Dimensionality Study of SSAM . . . . .	101

## LIST OF TABLES

1.1	Key differences between the two sequential methods. . . . .	17
3.1	Solutions quality breakdown for APL1P. . . . .	57
3.2	Summary of results for APL1P. . . . .	59
3.3	Summary of results for 10D. . . . .	60
3.4	Lower bound on sample size $N$ such that $\mathbb{P}(X_N^\delta \subset X^\varepsilon) \geq 1 - \alpha$ for 10D. . .	62
3.5	Change in sample size, stopping iteration, and solution time for APL1P . .	63
3.6	Summary of results for 20TERM . . . . .	65
5.1	Ratio of variance estimators for each pair of sampling scheme. . . . .	96
5.2	Empirical results on the asset allocation problem, where $h' = 0.5275$ . . . .	98
5.3	Characteristics of the derived instances from the 20TERM model . . . . .	99
5.4	Performance of SSAM as $d_\xi$ increases. $N_0 = 1000, h' = 1.500$ . . . . .	99

## ABSTRACT

Monte Carlo sampling-based methods are frequently used in stochastic programming when exact solution is not possible. In this dissertation, we develop two sets of Monte Carlo sampling-based algorithms to solve classes of two-stage stochastic programs. These algorithms follow a sequential framework such that a candidate solution is generated and evaluated at each step. If the solution is of desired quality, then the algorithm stops and outputs the candidate solution along with an approximate  $(1 - \alpha)$  confidence interval on its optimality gap.

The first set of algorithms proposed, which we refer to as the fixed-width sequential sampling methods, generate a candidate solution by solving a sampling approximation of the original problem. Using an independent sample, a confidence interval is built on the optimality gap of the candidate solution. The procedures stop when the confidence interval width plus an inflation factor falls below a pre-specified tolerance  $\epsilon$ . We present two variants. The fully sequential procedures use deterministic, nondecreasing sample size schedules, whereas in another variant, the sample size at the next iteration is determined using current statistical estimates. We establish desired asymptotic properties and present computational results.

In another set of sequential algorithms, we combine deterministically valid and sampling-based bounds. These algorithms, labeled sampling-based sequential approximation methods, take advantage of certain characteristics of the models such as convexity to generate candidate solutions and deterministic lower bounds through Jensen's inequality. A point estimate on the optimality gap is calculated by generating an upper bound through sampling. The procedure stops when the point estimate on the optimality gap falls below a fraction of its sample standard deviation. We show asymptotically that this algorithm finds a solution with a desired quality tolerance. We present variance reduction techniques and show their effectiveness through an empirical study.

## CHAPTER 1

## INTRODUCTION

Many real-life optimization problems include some element of uncertainty. Failure to properly consider uncertainty in decision making can result in under-performing if not disastrous decisions; for instance, by failing to consider unexpected events such as natural disasters. There are many approaches to account for uncertainty in decision making. Stochastic programming incorporates random variables and probabilistic statements into deterministic mathematical programming. In this setting, uncertainty is represented in the models through the use of random variables. Within the stochastic programming framework, in this dissertation, we consider models of the form:

$$z^* = \min_{x \in X} \{f(x) \equiv \mathbb{E}[F(x, \xi)]\}, \quad (\text{SP})$$

where  $X \subseteq \mathbb{R}^{d_x}$  is the set of constraints on the decision variables  $x$ ,  $\xi$  is a random vector with finite dimension  $d_\xi$  and known distribution over support  $\Xi \subseteq \mathbb{R}^{d_\xi}$ . Note that  $X$  does not contain any stochastic constraints such as those found in chance-constrained stochastic programs. The real-valued function of decisions  $x$  and a realization of the random vector  $\xi$  is denoted by  $F$ , and  $\mathbb{E}$  is the expectation operator. It is assumed that expectation of  $F(x, \xi)$ , denoted  $f(x)$ , is well-defined for all  $x \in X$ .

A variety of problems from different fields and applications can be formulated as (SP2). Financial planning, airline crew scheduling, production planning, energy planning, machine learning, and simulation are some examples. For instance, in machine learning, the function  $F(x, \xi)$  may represent the indicator function on a misclassification in a binary classification problem. In this case, the objective in solving (SP2) corresponds to minimizing the probability of misclassification. (SP2) also corresponds to many problems in simulation. For example, in simulation of queuing systems, one might be interested in minimizing the mean waiting time of the entities in the queue.

In this dissertation, we focus on a class of (SP2) known as the *two-stage stochastic programs with recourse*, which can be formulated as follows:

$$z^* = \min_{x \in X} \{f(x) \equiv c(x) + \mathbb{E}[Q(x, \xi)]\}, \quad (\text{SP2})$$

where

$$\begin{aligned} Q(x, \xi) = \min_{y \geq 0} \quad & q(y, \xi) \\ \text{s.t.} \quad & g(y, \xi) \leq h(\xi) - T(x, \xi). \end{aligned} \quad (1.1)$$

In this formulation,  $X$  is the set of the first-stage decisions,  $y \in \mathbb{R}^{d_y}$  represent the second-stage decisions,  $c : X \rightarrow \mathbb{R}$ ,  $Q : X \times \text{co}(\Xi) \rightarrow \mathbb{R}$ ,  $q : \mathbb{R}^{d_y} \times \text{co}(\Xi) \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^{d_y} \times \text{co}(\Xi) \rightarrow \mathbb{R}^{m_y}$ ,  $h : \text{co}(\Xi) \rightarrow \mathbb{R}^{m_y}$ , and  $T : X \times \text{co}(\Xi) \rightarrow \mathbb{R}^{m_y}$ , where  $\text{co}(\cdot)$  denotes the convex hull operator.

The above formulation encompasses both two-stage stochastic linear and nonlinear programs. Furthermore,  $X$  can contain integrality constraints. It is also possible to add integrality constraints to the second-stage decisions  $y$ . A subset of (SP2) includes the two-stage stochastic linear programs with recourse, in which all functions in  $x$  and  $y$  are linear and  $X = \{x : Ax = b, x \geq 0\}$ . This class of problems was introduced by Beale (1955) and Dantzig (1955).

In a recourse model, decisions  $x$  are made in the first stage before the random data  $\xi$  is observed, then *recourse* actions  $y$  are taken after observing the random data in the second stage. Examples of (SP2) can be found from many fields in the literature: telecommunications (Sen et al., 1994), transportation (Frantzeskakis and Powell, 1990), finance (Cariño et al., 1994), electricity power generation (Cariño et al., 1994), manufacturing (Eppen et al., 1989), and military applications (Morton et al., 1996; Baker et al., 2002). Below we will provide three examples of problems that can be modeled as two-stage stochastic programs with recourse. The first example is a static stochastic knapsack model, the second example is an electric power generation model, and the last example is a motor freight scheduling model. Instances of these problems are used for the computational results presented in later chapters of this dissertation.

**Example 1.1 (Static Stochastic Knapsack).** Consider a knapsack with a known capacity  $q$ , and a set of  $m$  items indexed by  $i = 1, 2, \dots, m$ . Suppose each item  $i$  has a known reward  $r_i$  and an unknown weight  $w_i$ . The uncertainty in the weights can be modeled by a known probability distribution with  $w_i$  independent of one another. A decision-maker wishes to load the knapsack with a selection of items from the set of  $m$  items provided so that the total rewards of the items in the knapsack is maximized. The decision-maker does not know the actual weights but only knows the distribution of the weights before selection. If the selection results in a total weight exceeding the capacity of the knapsack, a penalty cost of  $c$  per unit of capacity exceeded is assessed. The objective is to select the items with maximal expected net benefit (total rewards minus total cost). Detailed formulation of this problem along with related models and instances are provided by Ahmed and Shapiro (2002). In our computations, we use an instance referred to as 10D from Kleywegt et al. (2001).

**Example 1.2 (Electric Power Generation).** A power company can build up to two generators at a power plant to provide electricity for one region with three load levels: base, medium, and peak. The demands  $d_j$  associated with each load level ( $j = 1, 2, 3$ ) are independent random variables with known distributions. The plant manager needs to determine the capacity of each generator in order to meet demands. Once the generators are built, their actual operating capacities are random. There is a fixed cost  $f_i$  associated with building each generator  $i$  and an associated unit operating cost  $f_{ij}$  when generator  $i$  is running to meet one unit of demand of load level  $j$ . When the actual operating capacities fail to meet the demand at load level  $j$ , additional capacities can be purchased for a penalty cost of  $c_j > f_{ij}, \forall i$ . The goal is to determine what capacity to build the generators in order to minimize expected total cost (investment and operating costs). This model was proposed by Infanger (1992) and is referred to as the test problem APL1P.

**Example 1.3 (Motor Freight Scheduling).** Consider a motor freight carrier with a fleet of vehicles (e.g., tractors and trailers) to carry shipments (customer demands) to-and-fro a central terminal, local terminals, and other customer shipment centers. The operations of the motor freight carrier are conducted as follows: First, the manager of the carrier

*decides at which terminals to position a fleet of vehicles at the beginning of the day while customer shipment demands are unknown. Then, once demands are observed, decisions need to be made to move the fleet through a multi-commodity network in order to satisfy shipment demands in such a way that all the vehicles return back to their initial positions at the end of the day. Description of this model is provided by Mak et al. (1999). We use an instance of this problem denoted as 20TERM in our computational tests.*

### 1.1 Motivation

Unless the random vector  $\xi$  is low-dimensional,  $|\Xi|$  is of manageable size, or  $F$  has a special form that allows for quick solutions, (SP2) models of practical size are typically very difficult, if not impossible, to solve exactly. For a given decision vector  $x \in X$ , it is often the case that  $f(x)$  cannot be evaluated exactly due to the presence of the expectation operator in the objective function—this often requires the ability to calculate a high-dimensional integral. For example, we will consider one instance of the model described in Example 1.3, called 20TERM, where the random vector  $\xi$  has 40 random components with two realizations each. This instance results in a linear optimization problem with  $8.4 \times 10^{14}$  variables and  $1.4 \times 10^{14}$  constraints. To date, no algorithm has been designed that solves 20TERM to optimality.

Generally speaking, two types of algorithmic approaches exist to solve stochastic problems of the form of (SP2): the exact solution approach and the approximation approach. Exact-solution algorithms attempt to solve the deterministic equivalent directly. That is, they explicitly solve the model that results from including all of the realizations of the random vector  $\xi$  and writing out the expected value in its extensive form when the random vector has a discrete distribution with a finite support. Some of these algorithms include decomposition algorithms, such as the L-shaped method (Bender’s decomposition) that exploit the structure of two-stage stochastic linear programs and decompose them by each realization in order to solve them more efficiently. However, these algorithms can become quickly intractable when  $|\Xi|$  is very large (such as in the case of 20TERM) or  $|\Xi|$  is infinity (when  $\xi$  is continuous). Approximation methods, on the other

hand, solve an approximate problem of (SP2) that is computationally easier.

A commonly used approximation method consists of estimating the expectation by the sample mean and solving the resulting sampling problem for a given sample size  $n$ :

$$z_n^* = \min_{x \in X} \sum_{i=1}^n \frac{1}{n} F(x, \xi^i), \quad (\text{SP}_n)$$

where the observations  $\xi^1, \xi^2, \dots, \xi^n$  may form a random sample or they can be generated by a different type of sampling scheme. In Monte Carlo sampling average approximation,  $(\text{SP}_n)$  becomes tractable due to the reduction in the number of realizations considered. Let  $x_n^*$  denote an optimal solution to  $(\text{SP}_n)$ . The asymptotic properties of  $z_n^*$  and  $x_n^*$  have been extensively studied in the literature (Attouch and Wets, 1981; Dupačová and Wets, 1988; King and Rockafellar, 1993; Shapiro, 1989). However, in practice, only a finite sample size can be used. In this case, how large should the sample size be in order to obtain near-optimal solutions? Can we envision an algorithm that can generate candidate solutions by solving problems of the form of  $(\text{SP}_n)$  while making a valid statement about the quality of the solution output? These questions motivate the research presented in this dissertation. We also blend sampling approximations with other approximations that result in deterministic bounds on  $z^*$ , as described below.

Another classical approximation approach consists in solving an approximate problem whose optimal objective function value results in a deterministic bound, upper or lower, on  $z^*$ . Such approximate problems can be formulated by exploiting certain properties of the original model (SP2). For instance, when  $Q(x, \cdot)$  is convex on  $\Xi$ , we can apply Jensen's inequality to obtain a deterministic lower bound on  $z^*$  (for a minimization problem). Under some additional conditions on the support of the random vector, Mandasky's inequality can be applied to obtain a deterministic upper bound. While the deterministic bounds provide the certainty that the solutions of a sampling approximation lack, we still face the question of how tight the derived bounds are. When both deterministic lower and upper bounds are derived simultaneously for the same problem, then an interval around  $z^*$  can be constructed. These bounds can be sequentially refined, resulting in algorithms referred to as the *Sequential Approximation Methods* (Edirisinghe and Ziemba, 1992; Frauendorfer, 1992; Huang et al., 1977; Kall et al., 1988). Jensen's (lower) bound

is relatively easy to calculate but upper bounds can be computationally expensive. We therefore replace this upper bound with a cheaper Monte Carlo sampling-based bound.

## 1.2 Contributions

The goal of this dissertation is to use Monte Carlo sampling and deterministic bounding approaches within an algorithmic framework to generate approximate solutions to (SP2), along with an asymptotically valid statement on the quality of the solutions found. We note that there is a rich history of Monte Carlo sampling-based as well as deterministically-valid bounding algorithms for stochastic programming and we review these in the literature review sections of Chapters 2 and 4 (see §2.1 and §4.1). For many of the sampling-based methods, however, stopping criteria that *control* the quality of obtained solutions have largely gone unaddressed in the literature. While the sequential nature of sampling has been well studied in statistics and simulation, in stochastic programming, either a fixed sample size is used—which is analogous to running a deterministic optimization algorithm a pre-specified number of iterations—or, an adaptive but heuristic stopping rule is employed, ignoring the inherent sequential aspects associated with sampling. The work presented in this dissertation explicitly considers and analyzes theory to develop stopping rules that ensure a desired quality of the solutions obtained for these approximation algorithms. The main contribution of this dissertation is twofold:

1. In the sampling approximation case, we investigate stopping rules that provide sound theoretical guidelines on when the sample size is large enough to generate solutions to (SP2) that are of desired quality. The stopping rules are based on a confidence interval (CI) formed on the optimality gap of a candidate solution. A sequence of candidate solutions are generated by solving  $(SP_n)$  with increasing sample size and the procedure stops when the width of the CI on the optimality gap falls below a fixed desired width. We call this algorithm *Fixed-Width Sequential Sampling Method*.
2. In the deterministic bounding case, we develop an algorithm that makes use of the relatively easy Jensen’s lower bound and a sampling upper bound to find approxi-

	<b>Fixed-Width Sequential Sampling Method</b>	<b>Sampling-Based Sequential Approximation Method</b>
<b>Step 1</b> generate $\hat{x}_k$	Solve a Monte Carlo sampling approximation problem	Solve a lower bounding problem
<b>Step 2</b> assess quality	Monte Carlo statistical bounds CI on the optimality gap	Deterministic lower bound Statistical upper bound
<b>Step 3</b> stopping rule	Fixed-width stopping rule	Relative-width stopping rule
<b>Step 4</b> improve	Increase sample size	Refine Partition; Increase sample size

Table 1.1: Key differences between the two sequential methods.

mate solutions to (SP2) while making statements about the quality of the solutions found. We call this algorithm *Sampling-Based Sequential Approximation Method* because it is inspired by both the sequential sampling and the classical sequential approximation methods.

Both sequential methods developed in this dissertation follow the same algorithmic framework. The algorithms start by initializing necessary variables and parameters (step 0: initialization). A candidate solution at iteration  $k$ , denoted  $\hat{x}_k$ , is then generated (step 1: generate  $\hat{x}_k$ ). The quality of this solution is measured (step 2: assess quality) and the algorithms stop if the solution is of desired quality (step 3: stopping rule). Otherwise, the current solution is improved and re-evaluated until the desired quality is achieved (step 4: improve). Though both methods proposed follow the same algorithmic framework, their respective implementation of each step is different. In Table 1.1, we summarize the key differences between the two methods for each step. We elaborate more on these differences in later chapters.

### 1.3 Definitions and Assumptions

In this section, we introduce several terms and their relevant notations, which are used throughout the dissertation. We also list some additional assumptions on the class of (SP2) considered in this dissertation. We begin with the definition of optimality gap.

Consider a feasible decision vector  $\hat{x} \in X$ . We measure the quality of  $\hat{x}$  by its *optimality*

gap, denoted by  $\mu_{\hat{x}}$ , defined as the difference between the value of the objective function evaluated at  $\hat{x}$  and  $z^*$ :

$$\mu_{\hat{x}} = f(\hat{x}) - z^* = \mathbb{E}[F(\hat{x}, \xi)] - z^*. \quad (1.2)$$

The lower the  $\mu_{\hat{x}}$ , the higher the quality, and a zero  $\mu_{\hat{x}}$  implies  $\hat{x}$  is an optimal solution. Note that  $\mu_{\hat{x}} \geq 0$  for all  $x \in X$ . It is sometimes more convenient to work with the relative optimality gap instead (i.e., the optimality gap normalized with respect to  $z^*$ ) in which case we will say of the feasible decision vector  $\hat{x}$  that it is within  $(\mu_{\hat{x}}/z^*) \times 100\%$  of optimality.

Note that  $\mu_{\hat{x}}$  is the expected value of the random variable “ $F(\hat{x}, \xi) - z^*$ ”. In cases where  $\mu_{\hat{x}}$  cannot be computed exactly, given a random sample of size  $n$ , one can build a confidence interval  $I_n$  to estimate the range of values that contain  $\mu_{\hat{x}}$ . The *coverage probability* of the confidence interval  $I_n$  is defined as  $\mathbb{P}(\mu_{\hat{x}} \in I_n)$ . When the confidence interval has the desired coverage probability (e.g. 90%), it is called a *valid confidence interval*.

Let  $\omega = \{\xi^1, \xi^2, \dots\}$  be a sequence of independent realizations of the random vector  $\xi$ . When we say that *an event happens with probability one (w.p.1) for  $n$  large enough*, we mean that for almost every  $\omega$  there exists an integer  $N(\omega)$  such that the event happens for all samples of size  $n \geq N(\omega)$  from  $\omega$ .

Throughout this dissertation, we use the term *scenario* to designate a specific realization of the random vector  $\xi$ . Within the context of this terminology, an  *$n$ -scenario problem* (SP $_n$ ) is one where  $n$  realizations of  $\xi$  are used in the (SP2) model.

(SP2) is said to have *relatively completely recourse* if the optimization problem (1.1) has a finite optimal solution achieved by a feasible  $y$  for every element of  $X \times \text{co}(\Xi)$ . In what follows, we summarize general assumptions for the class of (SP2) considered. Assumptions specific to each of the algorithms proposed are introduced in later chapters, where these algorithms are developed in detail. The class of problems considered in this dissertation are of the form (SP2) and satisfy the following:

- (A1) (SP2) has relatively complete recourse;
- (A2)  $\mathbb{E}[Q(x, \xi)]$  is finite for all  $x \in X$ ; and,

(A3) (SP2) has a finite optimal solution in  $X$ .

#### 1.4 Dissertation Organization

This dissertation contains six chapters and the remaining five chapters are organized as follows. In Chapter 2, we present the *Fixed-Width Sequential Sampling Method*. We start this chapter by providing a review of relevant literature and discussing the problem class. We then review procedures developed earlier to build confidence intervals on the optimality gap of a given candidate solution, as these procedures are used in the sequential sampling methods presented. We present two versions of this algorithm. One version increases the sample sizes according to a fixed schedule, and the other estimates the next sample size using the current statistical estimates. We establish desired theoretical results for both versions.

Chapter 3 presents the empirical performance of the fixed-width sequential sampling methods. We also discuss implementation issues and provide guidelines for implementation.

In Chapter 4, we present the *Sampling-Based Sequential Approximation Method*. The chapter starts with a review of relevant literature and lists further assumptions on the class of problems considered. We then provide a review of the classical sequential approximation methods. We describe the proposed methods and establish key theoretical results.

Chapter 5 reports on the empirical performance of the sampling-based sequential approximation method. We analyze key settings and provide guidelines for successful application.

Finally, we conclude the dissertation in Chapter 6 by summarizing the contributions of this research and discussing future research directions.

## CHAPTER 2

## FIXED-WIDTH SEQUENTIAL SAMPLING METHOD (FWSSM)

In sequential Monte Carlo sampling-based approximation methods, a sequence of sampling problems of the form of  $(SP_n)$  are solved with an increasing sample size  $n$ . In order for these methods to be viable, we must be able to determine when to stop sampling to obtain solutions of desired quality. In this chapter, we investigate stopping rules for sequential sampling procedures that use confidence interval estimators on the optimality gap. We propose two procedures that attempt to find  $\varepsilon$ -optimal solutions to (SP2) as follows. The procedures generate candidate solutions by solving a sequence of sampling problems  $(SP_n)$  with increasing sample sizes and stop when the inflated width of the confidence interval on the optimality gap of the current solution is less than  $\varepsilon$ .

This chapter is organized as follows. In the next section, we review relevant literature to sequential sampling and related procedures. In §2.2, we discuss additional assumptions on the class of problems considered. In §2.3, we present a succinct review of the approaches used to build confidence intervals on the optimality gap of the candidate solutions. In §2.4, we review and establish relevant results for the limiting behavior of a sequence of random variables with a random index; these results are used in proving key theoretical properties of the algorithms proposed. In §2.5, we develop the fully sequential procedure (FSP) where the sample size schedules are deterministic. We show that FSP asymptotically finds an  $\varepsilon$ -optimal solution with probability one in a finite number of iterations and using finite sample sizes. In §2.6, we develop a variant of FSP for finding  $\varepsilon$ -optimal solutions where the sample size increases are random jumps based on current statistical estimates. We close this chapter with a summary in §2.7.

## 2.1 Literature Review

The motivation to study fixed-width sequential sampling methods comes from determining an adequate sample size to obtain high-quality solutions. Theoretical sample size bounds to obtain optimal solutions to (SP2) by solving sampling approximations ( $SP_n$ ) with a desired probability based on large deviations theory have been studied in the literature. Shapiro (2003) provides a summary of such theoretical sample size estimates. While these estimates provide insight into the complexity of solving (SP2) via Monte Carlo sampling approximation, they are, in general, much larger than what the empirical tests suggest (Verweij et al., 2003).

In this chapter, we approach the problem of finding a minimal sample size to reach a desired accuracy in a more algorithmic fashion. We consider sequential sampling procedures and mainly focus on *external sampling*, where a sampling problem with increasing sample size is solved to generate candidate solutions. We provide stopping rules and guidelines on how to increase the sample sizes such that these procedures stop at the minimal sample size, finding solutions within a desired accuracy. Our measure of “accuracy” is defined as being sufficiently close to the optimal objective function value. We measure this through a confidence interval formed on the optimality gap of the current candidate solution. The procedures stop when the width of the confidence interval on the optimality gap plus an inflation factor falls below a pre-specified, fixed value,  $\varepsilon$ . We show that the procedures stop w.p.1 for a given  $\varepsilon > 0$  and asymptotically find  $\varepsilon$ -optimal solutions to (SP), w.p.1. Our preliminary computational results indicate that the sample sizes when the procedures stop are far smaller than the theoretical estimates, while still providing high-quality solutions with high probability (see Chapter 3).

In the literature, various methods for building confidence intervals on the optimality gaps of candidate solutions have been developed (Bayraksan and Morton, 2006; Mak et al., 1999; Norikin et al., 1998). For the sequential sampling methods presented in this dissertation, we use the methods for forming optimality gap confidence intervals developed by Bayraksan and Morton (2006). Note that these methods are *static* by nature; i.e., a confidence interval on the optimality gap  $\mu_{\hat{x}}$  is formed for a given candidate solution

$\hat{x}$  and a fixed sample size  $n$ . In contrast, in our setting, the confidence intervals are used in an algorithmic framework, where both the candidate solution  $\hat{x}$  and the sample size  $n$  vary as the procedures progress. Thus, when the procedures stop, the sample size and the candidate solution at stoppage are both random variables. Consequently, the analysis of sequential methods require a much different treatment than their static counterparts.

There is an extensive literature on sequential analyses in statistics (Chow and Robbins, 1965; Ghosh et al., 1997; Nadas, 1969) and in simulation of stochastic systems (Glynn and Whitt, 1992; Hong and Nelson, 2005; Kim and Nelson, 2006; Law and Kelton, 1982; Law et al., 1981). A number of stopping rules have been proposed in the literature for specific sampling-based methods for stochastic programs (Dantzig and Infanger, 1995; Higle and Sen, 1991a, 1996a; Shapiro and Nemirovski, 2005; Norkin et al., 1998). However, except in the case of the stochastic quasi-gradient (SQG) methods, stopping rules that guarantee the quality of the candidate solutions found and their sequential nature have not been fully investigated. For the SQG methods, gradients and subgradients are estimated through sampling within a steepest descent approach. A survey on stopping rules and step sizes for SQG methods is provided by Pflug (1988). SQG methods can handle more general classes of problems, however they fail to take advantage of the underlying structure of the models. The procedures we develop in this chapter are based on sampling average approximations and can take advantage of efficient solution methods readily available specific to the models being studied.

Several researchers studied conditions on the growth of sample sizes such that certain desired properties are satisfied. For instance, Polak and Royset (2008) provide sample size estimates that minimize the computational effort spent in solving a certain class of stochastic nonlinear programs. In the same spirit, Futschik and Pflug (1997) develop a sampling strategy whose goal is to minimize the computational costs of solving stochastic programs through Monte Carlo sampling-based approximations. Homem-de-Mello (2003) derive conditions on the sample sizes in order to guarantee the consistency of the sampling-based estimators of  $z^*$ . Pasupathy (2010) derives growth rates for the sample sizes with respect to the sequence of error tolerances to ensure optimal convergence of the

solutions of retrospective-approximation algorithms for stochastic root finding and simulation optimization. While the results of Pasupathy (2010) are primarily concerned with the efficient convergence of the solutions obtained through sampling ( $x_n^*$ ) to an optimal solution ( $x^*$ ), our procedures are indifferent to how close a candidate solution is to the set of optimal solutions  $X^*$  but focus on whether the optimality gap of a candidate solution ( $\hat{x}$ ) is small enough ( $\leq \epsilon$ ).

Similar to our goal, Morton (1998) develops stopping rules theory and requirements for minimal growth in the sample sizes for a class of sampling-based algorithms with asymptotically normal optimality gap estimators. The asymptotic normality requirement, however, is rather restrictive as it is not typically satisfied for stochastic programs. Bayraksan and Morton (2011) extend on this approach by removing the asymptotic normality assumption and allowing the use of sample variance estimates for the unknown variances in the formulation of the sample size growth conditions. The procedures developed in this chapter follow in the same line of research while establishing alternative stopping rules and sample size growth guidelines for a class of stochastic programs.

We note several differences between the stopping rules of this chapter and those of Bayraksan and Morton (2011). First, the stopping rules in Bayraksan and Morton (2011) use *point estimators* of the optimality gap and variance, whereas the stopping rules of this chapter focus on the *interval estimator* of the optimality gap. In Bayraksan and Morton (2011), the stopping criteria depend on the point estimator of the optimality gap falling below a pre-specified *fraction* of the sampling variance; hence are considered *relative-width*. In contrast, the stopping rules for the procedures developed in this chapter depend on the *width* of the interval estimator falling below a pre-specified level,  $\epsilon$ , and are considered *fixed-width* (see §2.3 for details). The implication of this is that the relative-width stopping rules of Bayraksan and Morton (2011) provide a quality statement with respect to the sample variance estimator—if the variability is large, a worse quality statement is made. In contrast, the fixed-width stopping rules of this dissertation always aim to find an  $\epsilon$ -optimal solution. That is, they use a “fixed” quality statement. Note that the variability of the problem is typically captured in the sampling error part of an interval estimator, and this part shrinks to zero as the sample size increases even if the variability is large.

Another major difference is that, for the sequential procedures developed in this chapter, we mainly focus on finding the candidate solutions via solving a sampling problem with increasing sample size, whereas in Bayraksan and Morton (2011), the candidate solutions can be found by any method that eventually generates optimal solutions, w.p.1. We also focus on a more restrictive class of problems (see §2.2 for details) and sometimes focus on a specific class of interval estimators. These restrictions pay off in several ways. First, they allow us to obtain stronger results, e.g., asymptotic validity results with probability one instead of  $1 - \alpha$  as in Bayraksan and Morton (2011). Second, compared to Bayraksan and Morton (2011), we significantly increase our analysis on the growth of sample sizes. For instance, we analyze the growth needed in the sample sizes used for generating versus evaluating a candidate solution as well as the interplay between them such that the expected sample sizes used are finite. We examine the rate of increase in the sample sizes when the sequential procedures stop. Third, we present a way to increase the sample sizes in a stochastic way, by using the current estimates, in an effort to make the procedures more adaptive to the problem at hand.

## 2.2 Problem Class

Let  $X^*$  denote the set of optimal solutions to (SP2), and let  $x^* \in X^*$  be an optimal solution to (SP2) with optimal objective function value  $z^* = f(x^*)$ . Similarly, let  $X_n^*$  be the set of optimal solutions to (SP $_n$ ) with optimal objective function value  $z_n^*$ . Recall that we denote an optimal solution to (SP $_n$ ) by  $x_n^*$ . For the class of problems we consider, we make the following assumptions in addition to (A1)-(A3) from §1.3:

(A4) The event  $A_n = \{X_n^* \subseteq X^*\}$  happens w.p.1 for  $n$  large enough,

(A5)  $1 - P(A_n) \leq Ce^{-\beta n}$ , with  $C, \beta > 0, \forall n$ .

Assumption (A4) states that, for large enough sample sizes  $n$ , an optimal solution  $x_n^*$  to (SP $_n$ ) is also optimal to (SP2) w.p.1. According to (A5), this probability goes to one at an exponential rate. More details on these assumptions and the general class of problems that satisfy them are discussed by Shapiro (2003). For instance, if  $\mathbb{E}[\sup_{x \in X} |F(x, \xi)|] < \infty$ , then (A4) is satisfied when  $|\Xi| < \infty$ ,  $F(\cdot, \xi)$  is piecewise-linear convex for every  $\xi \in \Xi$ ,

$X$  is a closed convex polyhedral and  $X^* \neq \emptyset$  is bounded (Shapiro and Homem-de-Mello, 2000). Assumption (A4) is also satisfied when  $X$  is a finite set (Kleywegt et al., 2001). Stronger conditions are required for (A5) to be satisfied.

These conditions, and other results in stochastic programming, are often obtained from the application of large deviation theory (Kaniowski et al., 1995; Dai et al., 2000). It is not required that the sample be i.i.d in order for assumptions (A4) and (A5) to hold. In fact, Homem-de-Mello (2008) provides conditions under which (A4) and (A5) are satisfied under non i.i.d sampling. Non-i.i.d sampling may be desired for the purpose of variance reduction or to obtain a faster rate of convergence (larger  $\beta$ ) in (A5). For (SP2), assumptions (A4) and (A5) are satisfied when  $|\Xi|$  is finite,  $X$  is polyhedral (i.e., of the form  $X = \{x \in \mathbb{R}^{d_x} : Ax = b\}$ ),  $c(\cdot)$  and  $T(\cdot, \xi)$  are linear functions of  $x$  for a fixed  $\xi$ ,  $q(\cdot, \xi)$  and  $g(\cdot, \xi)$  are linear functions of  $y$  for a fixed  $\xi$ , and  $h(\xi)$  is linear on  $\Xi$ . Under these conditions, (SP2) is a two-stage linear program with recourse. When  $X \subset \mathbb{Z}^{d_x}$ , then we have a stochastic integer program, and (A4) and (A5) are satisfied under mild conditions (Kleywegt et al., 2001).

### 2.3 Confidence Interval Estimation of Optimality Gap

Recall that we measure the quality of a feasible solution  $x \in X$  to (SP2) by its optimality gap  $\mu_x = \mathbb{E}[F(x, \xi)] - z^*$ . For a sample of size  $n$ ,  $\xi^1, \xi^2, \dots, \xi^n$ , and a desired value for  $\alpha \in (0, 1)$ , suppose we have a mechanism to form a  $(1 - \alpha)$ -level confidence interval on  $\mu_x$ . We can write this confidence interval as follows:

$$[0, G_n(x) + v_n^\alpha(x)], \quad (2.1)$$

where  $G_n(x)$  represents a point estimator of the optimality gap, and  $v_n^\alpha(x)$  is the sampling error term. Since  $\mu_x \geq 0$  for all  $x \in X$ , we assume that  $G_n(x) \geq 0$  and  $v_n^\alpha(x) \geq 0$  w.p.1., and the confidence interval in (2.1) is a one-sided confidence interval bounded below by 0. The sequential algorithms we develop in this chapter use the confidence interval formed in (2.1) in the following manner. For a sequence of candidate solutions, we form confidence intervals on their optimality gaps of the form of (2.1). The procedure stops when the width of the confidence interval plus an inflation factor falls below a pre-specified value

$\varepsilon$ . This inflation function, which we will denote by  $h(n)$ , is a non-negative function of the sample size  $n$  and shrinks to zero as  $n \rightarrow \infty$ . We can consider, for instance,  $h(n) = 1/n$  or  $h(n) = 1/\sqrt{n}$ . The use of inflation functions in tightening the stopping criterion (here the width of the confidence interval on the optimality gap) is a very common technique in classical sequential statistical procedures (Chow and Robbins, 1965). Thus, the stopping criterion used for the procedures developed in this chapter can be written as

$$G_n(x) + v_n^\alpha(x) + h(n) \leq \varepsilon. \quad (2.2)$$

In words, the procedure stops at a candidate solution  $x$  using sample size  $n$  the first time inequality (2.2) is satisfied.

We consider a sequence of sample sizes  $n_k$  such that  $n_k \rightarrow \infty$  as  $k \rightarrow \infty$ . Furthermore, we make the following assumptions on the point estimator of the optimality gap and the sampling error term of the interval estimator:

$$(A6) \quad \lim_{k \rightarrow \infty} \mathbb{P}(\sup_{x \in X} |G_{n_k}(x) - \mu_x| > \delta) = 0 \text{ for any } \delta > 0,$$

$$(A7) \quad \text{Let } \alpha \in (0, 1) \text{ be given. Then, } \lim_{k \rightarrow \infty} \mathbb{P}(\sup_{x \in X} v_{n_k}^\alpha(x) > \delta) = 0, \text{ for any } \delta > 0.$$

According to assumption (A7), the sampling error term  $v_{n_k}^\alpha(x)$  must be defined such that it uniformly converges to zero in probability over  $X$ . Assumption (A6) requires an analogous uniform convergence property for the optimality gap estimator.

Bayraksan and Morton (2006) develop confidence interval estimators that satisfy (A6) and (A7) under appropriate conditions. We will use these confidence intervals in the sequential procedures developed in this chapter. In the following we briefly review the *single replication procedure (SRP)*, one of the confidence interval estimators proposed by Bayraksan and Morton (2006). Let  $\bar{f}_n(x) = \frac{1}{n} \sum_{i=1}^n F(x, \xi^i)$ . Note that, with this definition,  $z_n^* = \bar{f}_n(x_n^*)$ . For a given  $x \in X$ , SRP defines the point estimator on its optimality gap  $\mu_x$  and its associated sample variance as follows:

$$G_n(x) = \frac{1}{n} \sum_{i=1}^n F(x, \xi^i) - \min_{x \in X} \frac{1}{n} \sum_{i=1}^n F(x, \xi^i) = \bar{f}_n(x) - \bar{f}_n(x_n^*), \quad (2.3a)$$

$$s_n^2(x) = \frac{1}{n-1} \sum_{i=1}^n \left[ (F(x, \xi^i) - F(x_n^*, \xi^i)) - (\bar{f}_n(x) - \bar{f}_n(x_n^*)) \right]^2, \quad (2.3b)$$

where the same set of observations,  $\xi^1, \xi^2, \dots, \xi^n$ , are used in both terms in (2.3a) and the corresponding terms in (2.3b). Then, the optimality gap estimator in SRP, as defined in (2.3a), satisfies  $G_n(x) \geq 0$ , w.p.1. Given the optimality gap and variance estimators in (2.3a) and (2.3b), SRP forms a  $(1 - \alpha)$ -level confidence interval on  $\mu_x$  through

$$\left[ 0, G_n(x) + \frac{t_{n-1, \alpha} s_n(x)}{\sqrt{n}} \right], \quad (2.4)$$

where  $t_{n, \alpha}$  is the  $1 - \alpha$  quantile of the Student's  $t$  distribution with  $n$  degrees of freedom. Similarly, let  $z_\alpha$  be the  $1 - \alpha$  quantile of the standard normal. Under conditions provided by Bayraksan and Morton (2006), the point estimators in (2.3a) and (2.3b) are consistent (e.g.,  $G_n(x) \rightarrow \mu_x$ , w.p.1 as  $n \rightarrow \infty$  for all  $x \in X$ ), and the confidence interval in (2.4) is asymptotically valid, i.e.,

$$\liminf_{n \rightarrow \infty} \mathbb{P} \left( \mu_x \leq G_n(x) + \frac{z_\alpha s_n(x)}{\sqrt{n}} \right) \geq 1 - \alpha. \quad (2.5)$$

For some problems, SRP can perform poorly with small sample sizes, resulting in confidence intervals with low coverage probability. To improve coverage probability, variants of SRP where more than one replication are made can be used instead. Consider, for instance, the averaged 2-replication procedure (A2RP) where the sample size  $n$  is selected to be even and the sampled observations are divided into two random partitions. Gap and sample variance estimates,  $G_{n/2, i}(x)$  and  $s_{n/2, i}^2(x)$ , can be calculated as given in (2.3a) and (2.3b) for each partition  $i = 1, 2$ . The estimators from each partition are then pooled to obtain A2RP estimators:  $G_n(x) = \frac{1}{2}(G_{n/2, 1}(x) + G_{n/2, 2}(x))$  and  $s_n^2 = \frac{1}{2}(s_{n/2, 1}^2(x) + s_{n/2, 2}^2(x))$ . Using the pooled estimators, a confidence interval on the optimality gap  $\mu_x$  can then be formed as in (2.4). This approach of pooling estimators from more than one replication can be generalized for an arbitrary integer  $r \geq 1$  by selecting  $n$  to be a multiple of  $r$  and dividing the  $n$  observations into  $r$  random partitions. Then a confidence interval on the optimality gap can be built by pooling the  $r$  estimators. This approach can be generally denoted as the averaged  $r$ -replication procedure ( $A_r$ RP). In this notation, SRP corresponds to A1RP and we use the two interchangeably throughout this dissertation.

Consider a sequence of sample sizes  $\{m_k\}$  such that  $m_k \rightarrow \infty$  as  $k \rightarrow \infty$ . Suppose that a sequence of candidate solutions  $\{x_k\}$  are generated by solving the sequence of

sampling problems ( $SP_{m_k}$ ). If the confidence interval in (2.4) is formed according to  $ArRP$  ( $r \geq 1$ ), then uniform convergence of the sample means  $\bar{f}_{n_k}(x)$  to  $\mathbb{E}[F(x, \tilde{\xi})]$  on  $X$  w.p.1 is a sufficient condition for (A6) to be satisfied. When  $X$  is finite, for problems that satisfy (A4), this condition is typically already satisfied (Kleywegt et al., 2001). Thus, uniform convergence in this context virtually adds no more restriction on the class of problems considered. A sufficient condition for (A7) to be satisfied is that for some  $M > 0$ ,  $\lim_{n \rightarrow \infty} \mathbb{P}(\sup_{x \in X} s_n^2(x) < M) = 1$ . Bayraksan and Morton (2006) provide a set of requirements for this condition to hold.

The consistency and validity results such as the ones in (2.5) for  $ArRP$  are proven for a fixed  $x \in X$  as  $n \rightarrow \infty$ . This is in contrast to sequential procedures, such as the ones developed in this dissertation, that use these confidence intervals as stopping rules. For the sequential procedures, the candidate solution  $x_k$  as the procedure progresses changes with the random sample; and both the candidate solution and the sample size when the procedures stop are random variables. Therefore, the analysis of sequential sampling methods is more delicate. To this end, in the next section, we establish a number of theoretical results on the convergence of random variables with random indexes. These results will be used in later sections to show the validity and other key theoretical results of the procedures developed in this chapter.

## 2.4 Limit Theorems for Random Variables with Random Indexes

Consider a sequence of random variables  $\{Y_k, k = 1, 2, \dots\}$  and suppose the index  $k$  is replaced by a random index  $\{K(\varepsilon), \varepsilon \geq 0\}$  such that  $K(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Usually, these indices are defined as sample sizes but they can also represent other indices. For example,  $k$  could be the iteration number of a procedure and  $Y_k$  the random variable generated at iteration  $k$ , such as the candidate solution or a subgradient estimate generated at iteration  $k$ . Then, the random index  $K(\varepsilon)$  could be the *random* iteration number *when* the algorithm stops with respect to a stopping rule for a given  $\varepsilon$  and  $Y_{K(\varepsilon)}$  would then be the value of the random variable  $Y_k$  at iteration  $K(\varepsilon)$ . For a definition of random indices in the context our of sequential procedures, see §2.5.1 and §2.6.1. The result we review

below is concerned with convergence of the random variables as the random index tends to infinity.

**Theorem 1.** *Let  $\{Y_k, k = 1, 2, \dots\}$  be a sequence of random variables and  $\{K(\varepsilon), \varepsilon \geq 0\}$  be a family of positive, integer-valued random variables such that  $K(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Suppose*

$$Y_k \rightarrow Y, \text{ w.p.1 as } k \rightarrow \infty. \quad (2.6)$$

*Then,  $Y_{K(\varepsilon)} \rightarrow Y$ , w.p.1 as  $\varepsilon \downarrow 0$ .*

Theorem 1 is inspired by Richter (1965), see also Gut (1988). In Theorem 1, the random variables  $\{Y_k, k = 1, 2, \dots\}$  and the random index  $\{K(\varepsilon), \varepsilon \geq 0\}$  are not required to be independent. In fact, in many settings, including ours,  $K(\varepsilon)$  is a function of the random variables  $\{Y_k, k = 1, 2, \dots\}$ . Without independence, when convergence w.p.1 in (2.6) is replaced by convergence in probability, general limiting results for  $Y_{K(\varepsilon)}$  like above cannot be obtained, even when  $K(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$  (Gut, 1988). However, for a special case of (2.6) where  $Y_k = a$ , w.p.1 for  $k$  large enough where  $a \in \mathbb{R}$  is a constant, we can obtain a stronger result stated in the proposition below.

**Proposition 1.** *Let  $\{Y_k, k = 1, 2, \dots\}$  be a sequence of random variables and  $\{K(\varepsilon), \varepsilon \geq 0\}$  be a family of positive, integer-valued random variables such that  $K(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Let  $a \in \mathbb{R}$ . Suppose*

$$Y_k = a, \text{ w.p.1 for } k \text{ large enough.} \quad (2.7)$$

*Then,  $Y_{K(\varepsilon)} = a$ , w.p.1 for  $\varepsilon$  small enough.*

*Proof.* Our proof follows along the line of the proof of Theorem 1 provided by Richter (1965). Let  $A = \{\omega : Y_{K(\varepsilon, \omega)}(\omega) \neq a \text{ for } \varepsilon \text{ small enough and } K(\varepsilon, \omega) \rightarrow \infty \text{ as } \varepsilon \downarrow 0\}$ . It is sufficient to show that  $\mathbb{P}(A) = 0$ . To this end, select an element  $\omega$  from  $A$ . Then, for every  $k$ , one can find an  $\varepsilon \equiv \varepsilon(\omega)$  such that  $K(\varepsilon, \omega) \geq k$  and

$$|Y_{K(\varepsilon, \omega)}(\omega) - a| > 0. \quad (2.8)$$

By increasing  $k$ , one can obtain a sequence, which we denote by  $\{\varepsilon_t\}$  such that  $\varepsilon_t \downarrow 0$  as  $t \rightarrow \infty$ . Now, set  $k_t = K(\varepsilon_t, \omega)$ . Since  $\omega \in A$ ,  $k_t \rightarrow \infty$  as  $t \rightarrow \infty$ . Then, by (2.8),

$$|Y_{k_t}(\omega) - a| > 0,$$

for the sequence  $Y_{k_t}(\omega)$ . Hence,  $\omega$  is contained in the set  $B = \{\omega : Y_k \neq a \text{ for } k \text{ large enough and } K(\varepsilon) \rightarrow \infty \text{ as } \varepsilon \downarrow 0\}$ , i.e.,  $A \subseteq B$ . By hypothesis,  $\mathbb{P}(B) = 0$  and hence  $\mathbb{P}(A) = 0$  and the desired result follows.  $\square$

The result of Proposition 1 is quite intuitive: as soon as the index reaches the point where (2.7) is satisfied, the random nature of the index no longer has an effect. With these results established, we can now present our sequential procedures.

## 2.5 Fully Sequential Procedure

### 2.5.1 Description

In the sequential procedures developed in this chapter, two independent streams of observations are required: we use the stream  $\xi_1^1, \xi_1^2, \dots$  to generate a sequence of candidate solutions; and we use a second independent stream  $\xi_2^1, \xi_2^2, \dots$  to form confidence intervals on the optimality gap of the candidate solutions generated by the first stream. We denote the sequence of candidate solutions by  $\{\hat{x}_k\}$ . At iteration  $k$  of the algorithm, we use  $m_k$  observations from the first stream of random variables to solve  $(\text{SP}_{m_k})$ . The optimal solution to  $(\text{SP}_{m_k})$  is set as the candidate solution at iteration  $k$ , i.e.,  $\hat{x}_k = x_{m_k}^*$ . We use  $n_k$  observations from the second independent stream to calculate  $G_{n_k}(\hat{x}_k)$  and  $v_{n_k}^\alpha(\hat{x}_k)$  and form a CI on  $\mu_{\hat{x}_k}$  as in (2.4). The procedure stops when this CI plus an inflation factor,  $h(n_k)$ , falls below the pre-specified tolerance value  $\varepsilon \geq 0$ . At iteration  $k + 1$ , we have the option to either reuse the previously generated  $n_k$  observations and generate  $n_{k+1} - n_k$  new observations, or generate an entirely new set of  $n_{k+1}$  observations for the sampling problem  $(\text{SP}_{n_{k+1}})$ , whose solution gives the candidate solution at iteration  $k + 1$ . Similarly, we can either augment the previous  $m_k$  observations with  $m_{k+1} - m_k$  new observations or generate a new set of  $m_{k+1}$  observations to form the confidence interval on the optimality

gap. Augmentation can help reduce solution times when information from previous solutions can be used in solving the resulting sampling approximate problems (this is known as warm-starting). However, when only one sample path is utilized, the procedure can get stuck in a bad sample path (Homem-de-Mello, 2003) which may result in the procedure stopping too early at a solution of lower quality or requiring an excessively large number of iterations before stoppage at a desired solution. To avoid these possibilities, completely new set of observations can be generated from time to time (we use the term *resampling* to describe this approach). To this end, we define two new parameters as inputs to the procedures,  $k_f^m$  and  $k_f^n$ , denoting the *resampling frequency* for generating candidate solutions and for assessing solution quality, respectively. These parameters determine how often entirely new independent observations are generated and used for each stream. For instance, when  $k_f^n = 1$ , entirely new independent observations for assessing solution quality are generated at every iteration. On the other extreme,  $k_f^n = \infty$  corresponds to never generating entirely new set of independent observations, i.e., the samples are always augmented.

We develop two sequential sampling procedures within this framework: the fully sequential procedure (FSP) and the sequential procedure with stochastic schedules (SSP). We describe FSP in detail in this section. In FSP, sample size increases are made according to a deterministic schedule such that both  $m_k \rightarrow \infty$  and  $n_k \rightarrow \infty$  as  $k \rightarrow \infty$ . The fully sequential procedure is described as follows.

**FSP:**

**Input:** Sample size schedules  $\{m_k\}$  and  $\{n_k\}$ , inflation factor  $h(n)$ , desired values of  $\varepsilon > 0$ ,  $0 < \alpha < 1$ , and resampling frequencies  $k_f^n$  and  $k_f^m$ .

**Output:** A candidate solution,  $\hat{x}_{K(\varepsilon)}$ , and a confidence interval on its optimality gap with width of  $\varepsilon$ .

**Step 0.** (Initialization) Set  $k = 1$ , generate  $m_1$  observations  $\xi_1^1, \xi_1^2, \dots, \xi_1^{m_1}$  and independently generate  $n_1$  observations  $\xi_2^1, \xi_2^2, \dots, \xi_2^{n_1}$ .

**Step 1.** (Generate candidate solution) Using the observations  $\xi_1^1, \dots, \xi_1^{m_k}$ , solve  $(SP_{m_k})$  to obtain  $x_{m_k}^*$ . Set  $\hat{x}_k = x_{m_k}^*$ .

**Step 2.** (Assess solution quality) Using the observations  $\xi_2^1, \dots, \xi_2^{n_k}$ , calculate  $G_{n_k}(\hat{x}_k)$  and  $v_{n_k}^\alpha(\hat{x}_k)$ .

**Step 3.** (Check stopping criterion) If

$$G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_k) + h(n_k) \leq \varepsilon \quad (2.9)$$

then set  $K(\varepsilon) = k$ ,  $N(\varepsilon) = n_k$ ,  $M(\varepsilon) = m_k$ , and  $\hat{x}_{K(\varepsilon)} = \hat{x}_k$  and stop. Output candidate solution  $\hat{x}_{K(\varepsilon)}$  with confidence interval on  $\mu_{\hat{x}_{K(\varepsilon)}}$  as  $[0, \varepsilon]$ .

**Step 4.** (Improve: Increase sample size) If  $k_f^m$  divides  $k$  then sample  $\xi_1^1, \xi_1^2, \dots, \xi_1^{m_{k+1}}$  independently of samples generated in previous iterations. Else, sample  $m_{k+1} - m_k$  additional observations,  $\xi_1^{m_k+1}, \dots, \xi_1^{m_{k+1}}$ . Similarly, if  $k_f^n$  divides  $k$  then sample  $\xi_2^1, \xi_2^2, \dots, \xi_2^{n_{k+1}}$  independently of samples generated in previous iterations. Else, sample  $n_{k+1} - n_k$  additional observations,  $\xi_2^{n_k+1}, \dots, \xi_2^{n_{k+1}}$ . Set  $k = k + 1$  and go to 1.

If FSP stops according to (2.9), the iteration when the procedure stops,  $K(\varepsilon)$ , is a random variable defined by

$$K(\varepsilon) = \inf_{k \geq 1} \{k : G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_k) + h(n_k) \leq \varepsilon\}. \quad (2.10)$$

It is the first iteration when the inflated CI falls below  $\varepsilon$ . Later we provide conditions under which  $K(\varepsilon)$  is finite w.p.1. The sample sizes when the procedure stops,  $N(\varepsilon)$  and  $M(\varepsilon)$ , used to form the CI and to generate the candidate solution respectively, are then defined as  $N(\varepsilon) = n_{K(\varepsilon)}$  and  $M(\varepsilon) = m_{K(\varepsilon)}$ . As functions of  $K(\varepsilon)$ , both  $N(\varepsilon)$  and  $M(\varepsilon)$  are also random variables. The solution found by FSP and output by the algorithm is denoted by  $\hat{x}_{K(\varepsilon)}$ . Our goal in developing the algorithm is so that the candidate solution output,  $\hat{x}_{K(\varepsilon)}$ , has an optimality gap of at most  $\varepsilon$  with a desired  $(1 - \alpha)$  confidence, i.e.,  $\mathbb{P}(\mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon) \geq 1 - \alpha$ . For this result to be attainable, we first need a mechanism to find  $\varepsilon$ -optimal solutions. According to (A4),  $\varepsilon$ -optimal solutions can be obtained by solving the sampling approximate problem ( $\text{SP}_{M(\varepsilon)}$ ) if  $M(\varepsilon)$  grows sufficiently large. Next, due to the stopping rule (2.9), we need a mechanism to build valid confidence intervals for  $\varepsilon$ -optimal solutions with a high probability. For a valid confidence interval, i.e.,  $\mathbb{P}\left(\mu_{\hat{x}_{K(\varepsilon)}} \leq G_{N(\varepsilon)}(\hat{x}_{K(\varepsilon)}) + v_{N(\varepsilon)}^\alpha(\hat{x}_{K(\varepsilon)})\right) \geq 1 - \alpha$ , the stopping rule in (2.9) implies that  $\mathbb{P}\left(\mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon\right) \geq \mathbb{P}\left(\mu_{\hat{x}_{K(\varepsilon)}} \leq G_{N(\varepsilon)}(\hat{x}_{K(\varepsilon)}) + v_{N(\varepsilon)}^\alpha(\hat{x}_{K(\varepsilon)})\right) \geq 1 - \alpha$ , and we can achieve

the desired result of finding  $\varepsilon$ -optimal solutions with high probability. However, note that even in the case where the sample size,  $n$ , and the candidate solution,  $\hat{x}$ , are not random (as opposed to random variables  $N(\varepsilon)$  and  $\hat{x}_{K(\varepsilon)}$ ), results similar to those stated in (2.5) are shown only asymptotically, as  $n \rightarrow \infty$ . Therefore, we also need the sample size  $N(\varepsilon)$  to grow.

We add the inflation factor  $h(n_k)$  to the confidence interval width in the stopping rule so that the sample sizes,  $M(\varepsilon)$  and  $N(\varepsilon)$ , can grow arbitrarily large w.p.1 as  $\varepsilon \downarrow 0$ . Otherwise, when  $G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_k) = 0$ , the procedure stops at iteration  $k$  regardless of the value of  $\varepsilon > 0$ . For example, when SRP is used to form the confidence interval on the optimality gap, this can happen when the candidate solution found by solving  $(\text{SP}_{n_k})$ ,  $\hat{x}_k = x_{n_k}^*$ , is also an optimal solution for  $(\text{SP}_{m_k})$ . Then  $G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_k) = 0$  and, without the inflation factor, the procedure would stop even when  $\mu_{\hat{x}_k} \gg \varepsilon$ . To better understand the role of the inflation factor  $h(n)$ , suppose that the sample sizes for generating the candidate solution are increased one by one at every iteration, i.e.,  $m_k = n_k = n_0 + k - 1$  for some initial sample size  $n_0 \geq 2$ . Let  $h(n_k) = 1/\sqrt{n_k}$ . Then,  $\mathbb{P}(K(\varepsilon) \leq k) = 0$  for  $k \leq \lfloor \frac{1}{\varepsilon^2} \rfloor + 1 - n_0$ , where  $\lfloor \cdot \rfloor$  returns the largest integer less than or equal to its argument. Thus, the inflation factor allows  $K(\varepsilon)$ , hence  $N(\varepsilon)$  and  $M(\varepsilon)$ , to grow arbitrarily large, w.p.1 as  $\varepsilon \downarrow 0$ .

### 2.5.2 Asymptotic Validity, Finite Stopping, and Finite Sample Sizes

In this section, we establish three key theoretical properties of FSP. First, we provide conditions under which FSP finds an  $\varepsilon$ -optimal solution with probability one as  $\varepsilon \downarrow 0$ . Second, we guarantee, under certain assumptions, that the number of iterations and the sample sizes used when the procedure stops are finite w.p.1. Finally, we provide conditions under which the expected number of iterations and the expected sample sizes when the procedure stops are finite.

**Theorem 2.** *Assume (A4) holds. Let  $\{m_k\}$ ,  $\{n_k\}$  be positive, nondecreasing sequences such that  $m_k \rightarrow \infty$  and  $n_k \rightarrow \infty$  as  $k \rightarrow \infty$ . Let  $h(\cdot)$  be a nonnegative, nonincreasing function such that  $h(n_k) \rightarrow 0$  as  $n_k \rightarrow \infty$ , and suppose  $k < \infty$  implies that  $m_k < \infty$ ,  $n_k < \infty$ ,*

and  $h(n_k) > 0$ .

(i) Consider FSP that stops at iteration  $K(\varepsilon)$  according to stopping criterion (2.10), defined in terms of  $\varepsilon$ , using  $N(\varepsilon) = n_{K(\varepsilon)}$  and  $M(\varepsilon) = m_{K(\varepsilon)}$  samples. Then,

$$\lim_{\varepsilon \downarrow 0} \mathbb{P} \left( \mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon \right) = 1.$$

(ii) In addition, assume (A6) and (A7) hold. Given  $\varepsilon > 0$ ,

$$\mathbb{P}(K(\varepsilon) < \infty) = \mathbb{P}(N(\varepsilon) < \infty) = \mathbb{P}(M(\varepsilon) < \infty) = 1.$$

*Proof.* (i) Let  $Y_k = \mu_{\hat{x}_k} = E[f(\hat{x}_k, \tilde{\xi}) | \hat{x}_k] - z^*$ . Then, by (A4) and the fact that  $m_k \rightarrow \infty$  as  $k \rightarrow \infty$ ,  $Y_k = 0$ , w.p.1 for  $k$  large enough. The hypotheses of the theorem ensure  $K(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Therefore, using Proposition 1,  $Y_{K(\varepsilon)} = \mu_{\hat{x}_{K(\varepsilon)}} = 0$ , w.p.1 for  $\varepsilon$  small enough; hence, the desired result follows.

(ii) We first show that FSP stops in a finite number of iterations, w.p.1. Note that

$$\begin{aligned} \mathbb{P}(K(\varepsilon) = \infty) &\leq \lim_{k \rightarrow \infty} \mathbb{P} \left( G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_{n_k}) + h(n_k) > \varepsilon \right) \\ &\leq \lim_{k \rightarrow \infty} \mathbb{P} \left( |G_{n_k}(\hat{x}_k) - \mu_{\hat{x}_k}| + v_{n_k}^\alpha(\hat{x}_{n_k}) > \varepsilon - h(n_k) - \mu_{\hat{x}_k} \right) \\ &\leq \lim_{k \rightarrow \infty} \mathbb{P} \left( \sup_{x \in X} |G_{n_k}(x) - \mu_x| + \sup_{x \in X} v_{n_k}^\alpha(x) > \varepsilon - h(n_k) - \mu_{\hat{x}_k} \right) \\ &= 0, \end{aligned} \tag{2.11}$$

where (2.11) follows from (A4), (A6), (A7) and the fact that  $h(n_k)$  is shrinking to 0 as  $k$ , hence  $n_k$ , tends to infinity.  $\mathbb{P}(N(\varepsilon) < \infty) = 1$  and  $\mathbb{P}(M(\varepsilon) < \infty) = 1$  follow from the hypothesis that  $k < \infty$  implies  $n_k < \infty$  and  $m_k < \infty$ .  $\square$

In Theorem 2 part (i), we show that FSP finds an  $\varepsilon$ -optimal solution asymptotically, with probability one. We note that this is a much stronger result than the desired validity result  $\mathbb{P}(\mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon) \gtrsim 1 - \alpha$ . Though this result is asymptotic, i.e. as  $\varepsilon \downarrow 0$ , in practice we can expect  $\mathbb{P}(\mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon) \gtrsim 1 - \alpha$  for a fixed and small enough  $\varepsilon > 0$ . Note that, even in much simpler settings such as forming confidence intervals on the mean, validity results for sequential procedures are typically shown only asymptotically (Chow and Robbins, 1965). However, these sequential procedures have been successfully applied in various fields such as clinical statistical studies (Ghosh and Sen, 1991).

In part (ii) of Theorem 2, we show that, for a given  $\varepsilon > 0$ , FSP terminates in a finite number of iterations and uses a finite number of samples w.p.1. Unlike the results in part (i) of Theorem 2, these results are established for a *fixed*  $\varepsilon > 0$ . Note that  $\mathbb{P}(K(\varepsilon) < \infty) = 1$  does not necessarily imply  $\mathbb{E}[K(\varepsilon)] < \infty$ . However, under certain conditions, we can establish the finiteness of the expected value of the iteration number and the number of the samples used when the procedure stops. To show these results, we assume that (SP2) has a unique solution, i.e.,  $X^*$  is a singleton, and ArRP is used to form the confidence intervals on the optimality gap for  $r \geq 1$ . A minor additional requirement on the number  $n_k$  of samples used for assessing the solution quality is that  $n_k$  is a multiple of  $r$  when ArRP is used so that the  $n_k$  observations can be evenly divided among the  $r$  random partitions used to calculate the pooled gap estimates and sample variances. The proposition below summarizes the conditions on the schedule of sample sizes  $\{m_k\}$  and  $\{n_k\}$  under which the expected sample sizes and the expected number of iterations of FSP are finite for a given  $\varepsilon > 0$ .

**Proposition 2.** *Assume (A5) holds and (SP2) has a unique optimal solution,  $X^* = \{x^*\}$ . Let  $\{m_k\}$ ,  $\{n_k\}$  be positive, nondecreasing sequences such that  $m_k \rightarrow \infty$  and  $n_k \rightarrow \infty$  as  $k \rightarrow \infty$ . Let  $h(\cdot)$  be a nonnegative, nonincreasing function such that  $h(n_k) \rightarrow 0$  as  $n_k \rightarrow \infty$ , and suppose  $k < \infty$  implies that  $m_k < \infty$ ,  $n_k < \infty$ , and  $h(n_k) > 0$ . Consider FSP that uses ArRP,  $r \geq 1$ , based on SRP given in (2.3)-(2.4), to calculate the confidence intervals and stops at iteration  $K(\varepsilon)$  according to stopping criterion (2.10), defined in terms of  $\varepsilon$ , using  $N(\varepsilon) = n_{K(\varepsilon)}$  and  $M(\varepsilon) = m_{K(\varepsilon)}$  samples. Given  $\varepsilon > 0$ ,*

$$(i) \text{ if } \sum_{k=1}^{\infty} e^{-\beta m_k} < \infty \text{ and } \sum_{k=1}^{\infty} e^{-\beta n_k} < \infty \text{ for all } \beta > 0, \quad (2.12)$$

then,  $\mathbb{E}[K(\varepsilon)] < \infty$ ,

$$(ii) \text{ if } \sum_{k=1}^{\infty} (n_{k+1} - n_k) e^{-\beta m_k} < \infty \text{ and } \sum_{k=1}^{\infty} (n_{k+1} - n_k) e^{-\beta n_k} < \infty \text{ for all } \beta > 0, \quad (2.13)$$

then,  $\mathbb{E}[N(\varepsilon)] < \infty$ ,

$$(iii) \text{ if } \sum_{k=1}^{\infty} (m_{k+1} - m_k) e^{-\beta m_k} < \infty \text{ and } \sum_{k=1}^{\infty} (m_{k+1} - m_k) e^{-\beta n_k} < \infty \text{ for all } \beta > 0, \quad (2.14)$$

then,  $\mathbb{E}[M(\varepsilon)] < \infty$ .

*Proof.* (i) Let  $B_k$  denote the event  $\{G_{n_k}(\hat{x}_k) + v_{n_k}^\alpha(\hat{x}_k) + h(n_k) > \varepsilon\}$ , and let  $\{A\}^c$  denote the complement of event  $A$ . Since  $K(\varepsilon)$  is a nonnegative random variable,

$$\begin{aligned} \mathbb{E}[K(\varepsilon)] &= \sum_{k=0}^{\infty} \mathbb{P}(K(\varepsilon) > k) \\ &\leq 1 + \sum_{k=1}^{\infty} \mathbb{P}(B_k). \end{aligned} \quad (2.15)$$

Let  $k_0$  be the smallest positive integer such that  $h(n_k) \leq \varepsilon$  for all  $k \geq k_0$ . Note that since  $A_r$ RP is used to calculate the confidence intervals, for  $k \geq k_0$ ,

$$B_k \subseteq \{\hat{x}_k = x_{n_k/r,i}^*, \forall i = 1, \dots, r\}^c \subseteq \left\{ \{x_{m_k}^* \neq x^*\} \cup \left( \bigcup_{i=1}^r \{x_{n_k/r,i}^* \neq x^*\} \right) \right\}. \quad (2.16)$$

Otherwise, when  $\hat{x}_k = x_{n_k/r,i}^*$  for all  $i = 1, 2, \dots, r$ ,  $G_{n_k}(\hat{x}_k) = v_{n_k}^\alpha(\hat{x}_k) = 0$  and the condition for event  $B_k$  is not satisfied for  $k \geq k_0$ . Using the right-hand side of (2.16) and continuing from (2.15),

$$\begin{aligned} \mathbb{E}[K(\varepsilon)] &\leq k_0 + \sum_{k=k_0}^{\infty} P(x_{m_k}^* \neq x^*) + \sum_{i=1}^r \sum_{k=k_0}^{\infty} P(x_{n_k/r,i}^* \neq x^*) \\ &\leq k_0 + C \sum_{k=k_0}^{\infty} e^{-\beta m_k} + \sum_{i=1}^r \left( C \sum_{k=k_0}^{\infty} e^{-\beta n_k/r} \right) \end{aligned} \quad (2.17)$$

$$\begin{aligned} &= k_0 + C \sum_{k=k_0}^{\infty} e^{-\beta m_k} + rC \sum_{k=k_0}^{\infty} e^{-\beta n_k/r} \\ &< \infty, \end{aligned} \quad (2.18)$$

where (2.17) follows from (A5) and (2.18) follows from (2.12).

(ii) Let  $k_0$  be defined as above. Then,

$$\begin{aligned} \mathbb{E}[N(\varepsilon)] &= \sum_{n=0}^{\infty} \mathbb{P}(N(\varepsilon) > n) \\ &= n_{k_0} + \sum_{k=k_0}^{\infty} (n_{k+1} - n_k) \mathbb{P}(K(\varepsilon) > k) \end{aligned} \quad (2.19)$$

$$\leq n_{k_0} + C \sum_{k=k_0}^{\infty} (n_{k+1} - n_k) e^{-\beta m_k} + rC \sum_{k=k_0}^{\infty} (n_{k+1} - n_k) e^{-\beta n_k/r}, \quad (2.20)$$

where (2.19) follows from the fact that  $\mathbb{P}(N(\varepsilon) > n) = 1$  for  $0 \leq n \leq n_{k_0} - 1$ ;  $\mathbb{P}(N(\varepsilon) > n) = \mathbb{P}(K(\varepsilon) > k_0)$  for  $n_{k_0} \leq n \leq n_{k_0+1} - 1$ , etc. The right hand side of (2.20) is finite by (2.13). Proof of (iii) is analogous.  $\square$

Instances where the conditions in part (i) of Proposition 2 are satisfied include when the sample sizes  $m_k$  and  $n_k$  both grow of order  $O(\ln^2 k)$ ,  $O(k)$ , or higher such as  $O(k^2)$ , etc. These conditions would not be satisfied, however, if instead the sample sizes both grow of order  $O(\ln k)$ . To see this, consider the following example. Suppose SRP is used in Step 2 of FSP with the following schedules of sample sizes:  $m_k = n_k = \lceil c_0 + c_1 \ln^2 k \rceil$ , where  $c_0 \geq 2$  and  $c_1 > 0$  are two constants and  $\lceil \cdot \rceil$  returns the smallest integer greater than or equal to its argument. These guarantee  $\mathbb{E}[K(\varepsilon)] < \infty$ . If instead we use the schedules  $m_k = n_k = \lceil c_0 + c_1 \ln k \rceil$ , then the conditions in (2.12) are satisfied if and only if  $c_1 \beta > 1$ . This would require that we have reliable estimates for  $\beta$ . However, since finding such estimates is extremely difficult, we exclude such schedules from consideration here and state the conditions on the sample size schedules in Proposition 2 for all  $\beta > 0$ .

Conditions in parts (ii) and (iii) of Proposition 2 indicate that there needs to be a “balance” between the growths in sample sizes  $m_k$  and  $n_k$ . Consider for instance, when  $m_k$  is increased too fast relative to  $n_k$ . Even if FSP stops in a finite number of iterations w.p.1 and  $\mathbb{E}[K(\varepsilon)] < \infty$ , we cannot guarantee  $\mathbb{E}[M(\varepsilon)] < \infty$ . In this case, the procedure might fail to stop long after good solutions have been found as it might fail to detect those solutions early enough as good ones. As an example of a case where this might happen, consider the schedules  $m_k = k^k$  and  $n_k = \lceil 2 + \ln^2 k \rceil$ . Note that (2.14) is not satisfied. Suppose  $\beta = 1$ ,  $C = 1$  and  $\mathbb{P}(K(\varepsilon) > k) = e^{-3} k^{-\ln k}$  for  $k \geq 1$ . Then,  $\mathbb{E}[M(\varepsilon)] = \sum_{n=0}^{\infty} \mathbb{P}(M(\varepsilon) > n) = 1 + \sum_{k=1}^{\infty} ((k+1)^{k+1} - k^k) e^{-3} k^{-\ln k} = \infty$ . In this case we have  $\mathbb{E}[M(\varepsilon)] = \infty$ , even though FSP stops at the first iteration with a high probability and the expected number of iterations is only  $\mathbb{E}[K(\varepsilon)] = 1.111$ . Now, suppose we change the schedule  $\{n_k\}$  to  $n_1 = 2$  and  $n_k = k^k$ , for  $k \geq 2$ . In this case,  $n_k$  and  $m_k$  both grow of  $O(k^k)$  and (2.14) is satisfied. The difference is that  $\mathbb{P}(K(\varepsilon) > k)$  can no longer be  $e^{-3} k^{-\ln k}$  and has to be much smaller,  $\mathbb{P}(K(\varepsilon) > k) \leq 2e^{-k^k}$  for  $k \geq 2$ ; resulting in  $\mathbb{E}[M(\varepsilon)] < \infty$ .

### 2.5.3 Rate of increase of $M(\varepsilon)$ and $N(\varepsilon)$

In this section, we establish the rate of increase of  $M(\varepsilon)$  and  $N(\varepsilon)$  as  $\varepsilon \downarrow 0$  for certain schedules of sample sizes. Recall that while  $\{m_k\}$  and  $\{n_k\}$  are deterministic sequences, the sample sizes used when the procedure stops,  $M(\varepsilon)$  and  $N(\varepsilon)$ , are random variables for

a given  $\varepsilon > 0$ . For the following analysis, we set the inflation factor  $h(n) = 1/\sqrt{n}$  and for simplicity, set  $m_k = n_k$ , even though  $m_k$  could be a multiple of  $n_k$ . We consider a class of sample size schedules defined as a function of the iteration number,  $k$ , as follows:

$$m_k = n_k = \lceil c_0 + c_1 g(k) \rceil, \quad (2.21)$$

where  $c_0, c_1 > 0$  are two constants. We assume  $n_1 \geq 2$ . The growth function,  $g(k)$ , is a nonnegative, nondecreasing function of  $k$  satisfying the following two properties:

$$\lim_{k \rightarrow \infty} g(k) = \infty \quad \text{and} \quad \lim_{k \rightarrow \infty} (g(k) - g(k-1)) \leq U, \quad (2.22)$$

where  $U \in [0, \infty)$ . The first limit property in (2.22) guarantees that  $m_k \rightarrow \infty$  and  $n_k \rightarrow \infty$  as  $k \rightarrow \infty$ , while the second limit property ensures that the increase in sample sizes is finite as  $k \rightarrow \infty$ . Examples of functions that meet the requirements in (2.22) include linear and sub-linear functions such as  $g(k) = k$  and  $g(k) = \ln^2 k$ . A quadratic increase,  $g(k) = k^2$ , in contrast, does not satisfy the second limit condition in (2.22). In Proposition 4 below, we show that both  $N(\varepsilon)$  and  $M(\varepsilon)$  grow asymptotically at the same rate as  $1/\varepsilon^2$ , w.p.1 for sample size schedules defined according to (2.21) and (2.22). Before presenting Proposition 4, we introduce the following lemma whose results we will use in proving Proposition 4.

**Lemma 3.** *Assume (A4) holds and (SP2) has a unique optimal solution,  $X^* = \{x^*\}$ . Given  $n$  and  $x \in X$ , let  $G_n(x)$  and  $s_n^2(x)$  be calculated using ArRP,  $r \geq 1$  based on SRP given in (2.3). Furthermore, assume that  $\{M(\varepsilon)\}$  and  $\{N(\varepsilon)\}$  are independent sequences of positive, integer-valued random variables such that  $M(\varepsilon) \rightarrow \infty$  and  $N(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Let  $\hat{x}_\varepsilon \equiv x_{M(\varepsilon)}^*$  be obtained by solving a sampling problem with sample size  $M(\varepsilon)$ . Then,*

- (i)  $\lim_{\varepsilon \downarrow 0} \sqrt{N(\varepsilon)} G_{N(\varepsilon)}(\hat{x}_\varepsilon) = 0$ , w.p.1,
- (ii)  $\lim_{\varepsilon \downarrow 0} s_{N(\varepsilon)}(\hat{x}_\varepsilon) = 0$ , w.p.1.

*Proof.* We will show the result for SRP. The same result for ArRP,  $r \geq 2$  follows from the fact that ArRP estimators are formed by averaging  $r$  SRP estimators. Consider  $x_m^*$  and  $x_n^*$  obtained by solving sampling problems (SP <sub>$m$</sub> ) and (SP <sub>$n$</sub> ), respectively. As  $m \rightarrow \infty$  and

$n \rightarrow \infty$ , by (A4) and unique optimality,  $x_m^* = x^* = x_n^*$ , w.p.1 for  $n$  and  $m$  large enough. The hypotheses of the lemma states that  $M(\varepsilon) \rightarrow \infty$  and  $N(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Therefore, using a vector version of Proposition 1,  $\hat{x}_\varepsilon = x_{M(\varepsilon)}^* = x^* = x_{N(\varepsilon)}^*$ , w.p.1 for  $\varepsilon$  small enough. When this happens,  $G_{N(\varepsilon)}(\hat{x}_\varepsilon) = \bar{f}_{N(\varepsilon)}(\hat{x}_\varepsilon) - \bar{f}_{N(\varepsilon)}(x_{N(\varepsilon)}^*) = 0$ . Hence,  $G_{N(\varepsilon)}(\hat{x}_\varepsilon) = 0$ , w.p.1 for  $\varepsilon$  small enough. This implies (i). When  $\hat{x}_\varepsilon = x_{N(\varepsilon)}^*$ ,  $s_{N(\varepsilon)}^2(\hat{x}_\varepsilon) = 0$ , thus, proof of (ii) is essentially the same.  $\square$

When the class of schedules that satisfy (2.21) and (2.22) is used for FSP, it can be shown that  $M(\varepsilon)$  and  $N(\varepsilon)$  asymptotically increase at the rate of  $1/\varepsilon^2$ , w.p.1 under a unique optimality assumption. This result is formally expressed in the next proposition.

**Proposition 4.** *Assume (A4) holds and (SP2) has a unique optimal solution,  $X^* = \{x^*\}$ . Let  $\{m_k\}$ ,  $\{n_k\}$  be positive, integer-valued nondecreasing sequences that satisfy (2.21)-(2.22) and let  $h(n) = 1/\sqrt{n}$ . Consider FSP that uses ArRP,  $r \geq 1$ , based on SRP given in (2.3)-(2.4), to calculate the confidence intervals and stops at iteration  $K(\varepsilon)$  according to (2.10) using  $N(\varepsilon) = n_{K(\varepsilon)}$  and  $M(\varepsilon) = m_{K(\varepsilon)}$  samples. Then,*

$$\lim_{\varepsilon \downarrow 0} \frac{M(\varepsilon)}{1/\varepsilon^2} = 1, \quad \text{and} \quad \lim_{\varepsilon \downarrow 0} \frac{N(\varepsilon)}{1/\varepsilon^2} = 1, \quad \text{w.p.1.} \quad (2.23)$$

*Proof.* Let  $G_{n_k} = G_{n_k}(\hat{x}_k)$ ,  $s_{n_k} = s_{n_k}(\hat{x}_k)$ ,  $t_k = t_{n_k-1, \alpha}$ . When FSP stops, (2.9) is satisfied. Rearranging (2.9), we have

$$\frac{\left(\sqrt{N(\varepsilon)}G_{N(\varepsilon)} + t_{K(\varepsilon)}s_{N(\varepsilon)} + 1\right)^2}{\varepsilon^2} \leq N(\varepsilon). \quad (2.24)$$

Note that (2.9) is not satisfied at iteration  $K(\varepsilon) - 1$ . Rearranging the terms again, adding  $n_1$  to the right-hand side and noting that  $n_{K(\varepsilon)-1} \geq n_{K(\varepsilon)} - c_1 [g(K(\varepsilon)) - g(K(\varepsilon) - 1)] - 1$ , we obtain

$$N(\varepsilon) \leq n_1 + 1 + c_1 [g(K(\varepsilon)) - g(K(\varepsilon) - 1)] + \frac{\left(\sqrt{n_{K(\varepsilon)-1}}G_{n_{K(\varepsilon)-1}} + t_{K(\varepsilon)-1}s_{n_{K(\varepsilon)-1}} + 1\right)^2}{\varepsilon^2}, \quad (2.25)$$

where we set  $m_{K(\varepsilon)-1} = m_1$  and  $n_{K(\varepsilon)-1} = n_1$  when  $K(\varepsilon) = 1$ . The inequality (2.25) is satisfied since, if the procedure stops at iteration 1, i.e.,  $K(\varepsilon) = 1$ , then,  $N(\varepsilon) = n_1$ ,

otherwise, it is less than or equal to the sum of the last three terms on the right-hand side of (2.25). Dividing each term in (2.24) and (2.25) by  $1/\varepsilon^2$  and taking limits as  $\varepsilon \downarrow 0$ , we obtain,

$$1 \leq \lim_{\varepsilon \downarrow 0} \frac{N(\varepsilon)}{1/\varepsilon^2} \leq 1, \text{ w.p.1.} \quad (2.26)$$

In (2.26), the “1” on the left-hand side is obtained by invoking Lemma 3 and the fact that  $t_{K(\varepsilon)} \rightarrow z_\alpha$ , w.p.1 as  $\varepsilon \downarrow 0$ . The “1” on the right-hand side is derived by noting that as  $\varepsilon \downarrow 0$ : (i)  $\varepsilon^2(n_1 + 1 + c_1 [g(K(\varepsilon)) - g(K(\varepsilon) - 1)]) \rightarrow 0$ , w.p.1 since  $n_1 + 1$  is a constant and  $\lim_{\varepsilon \downarrow 0} c_1 [g(K(\varepsilon)) - g(K(\varepsilon) - 1)] < \infty$ , w.p.1, and (ii)  $K(\varepsilon) - 1 \rightarrow \infty$ ,  $m_{K(\varepsilon)-1} \rightarrow \infty$ , and  $n_{K(\varepsilon)-1} \rightarrow \infty$ ; hence, the numerator of the last term in (2.25) tends to 1 by again invoking Lemma 3. By (2.21), the same result holds for  $M(\varepsilon)$ .  $\square$

The implication of Proposition 4 can be explained as follows. In order to obtain higher-quality solutions using FSP, larger sample sizes with asymptotic increases of  $1/\varepsilon^2$  are needed. Sample size increases of order  $O(\varepsilon^{-2})$  have been observed for other Monte Carlo sampling-based algorithms. For stochastic discrete optimization, theoretical sample size bounds in order to obtain  $\varepsilon$ -optimal solutions with a desired probability via sampling-based approximations require similar increases (Kleywegt et al., 2001). Further examples of similar type of sample size increases are found in the stochastic approximation method (Nemirovski et al., 2009) and in classical sequential procedures in statistics (Ghosh et al., 1997).

#### 2.5.4 Parameter Values of Linear Schedules

We consider linear schedules where  $g(k) = k - 1$  so that the initial sample size is  $n_1 = \lceil c_0 \rceil$ . These schedules satisfy (2.22). The use of linear schedules is very popular in both sequential statistics and stochastic programming. For instance, the stochastic decomposition method (Higle and Sen, 1996b) uses a linear schedule where the sample size is increased one by one as it increases computational efficiency. For this class of schedules, Proposition 4 shows that, as  $\varepsilon \downarrow 0$ ,  $N(\varepsilon)$  grows of order  $O(\varepsilon^{-2})$ , w.p.1, under certain conditions. However, we know from Theorem 2 that for  $\varepsilon > 0$ ,  $\mathbb{P}(N(\varepsilon) < \infty) = 1$ ; and

moreover,  $\mathbb{E}[N(\varepsilon)] < \infty$  by Proposition 2. We can then try to determine the values of the constants  $c_0$  and  $c_1$  that minimize the expected sample sizes when the procedure stops. Since exact minimization of  $\mathbb{E}[N(\varepsilon)]$  (and consequently of  $\mathbb{E}[M(\varepsilon)]$ ) may not always be possible, we will instead minimize the upper bound provided in (2.20). For a given value of  $r$ , by setting  $k_0 = 1$  in (2.20) and ignoring the integrality requirement, we obtain:

$$\mathbb{E}[N(\varepsilon)] \leq u(c_0, c_1) = c_0 + C \frac{c_1}{1 - e^{-\beta c_1}} e^{-\beta c_0} + Cr \frac{c_1}{1 - e^{-\beta c_1/r}} e^{-\beta c_0/r}. \quad (2.27)$$

The upper bound  $u(c_0, c_1)$  in (2.27) has the following two properties:

1.  $u(c_0, \cdot)$  is an increasing function of  $c_1$  on  $(0, \infty)$
2.  $u(\cdot, c_1)$  is a convex function of  $c_0$  on  $(0, \infty)$ .

When  $ArRP$  is used, by the first property, we can set  $c_1 = r$  and try to minimize  $u(\cdot, r)$ . That is, we increase the sample sizes by  $r$  observations at each iteration. Then, we can determine the value of  $c_0$  that minimizes  $u(\cdot, r)$  by  $c_0^* = \{c_0 > 0 : \frac{\partial}{\partial c_0} u(c_0, r) = 0\}$ . In the case of FSP with A1RP,  $c_0^* = \frac{1}{\beta} \ln(\frac{2\beta C}{1 - e^{-\beta}})$ ; and in the case of FSP with A2RP,  $c_0^* = \frac{2}{\beta} \ln\left(\frac{2A}{-B + \sqrt{B^2 + 4A}}\right)$  where  $A = \frac{2\beta C}{1 - e^{-2\beta}}$  and  $B = \frac{2\beta C}{1 - e^{-\beta}}$ . We can then round up  $c_0^*$  to the nearest integer divisible by  $r$  to find the initial sample sizes.

Figure 2.1 shows the values of  $c_0^*$  as a function of  $\beta$  for both A1RP and A2RP. Note that  $c_0^*$  is a decreasing function of  $\beta$  so that smaller values of  $\beta$  correspond to larger values of  $c_0^*$ . Derived values for  $c_0^*$  are approximately twice as large for A2RP as they are for A1RP. To see a practical value of  $c_0^*$ , we experimented on an instance of the static stochastic knapsack example described in Example 1.1. This instance, denoted 10D (see Chapter 3 for details), contains 10 objects (with random weights) among which a selection has to be made to go into the knapsack. Thus, 10D has 10 decision variables with independent normally distributed stochastic parameters (Kleywegt et al., 2001). Exploiting the properties of the normal distribution, we derived an estimate for  $\beta$  that is  $\hat{\beta} = 9.9625 \times 10^{-8}$ . Using  $\hat{\beta}$ , we obtain  $\hat{c}_0 = 7.65 \times 10^7$  and  $\hat{c}_0 = 1.50 \times 10^8$  as estimates of  $c_0^*$  for A1RP and A2RP, respectively. Unfortunately, since  $u(c_0, c_1)$  is based on worst-case bounds, these initial sample size values are too large to be of practical value. They exceed by far the sample sizes utilized by FSP in our computational experiments (see Chapter 3).

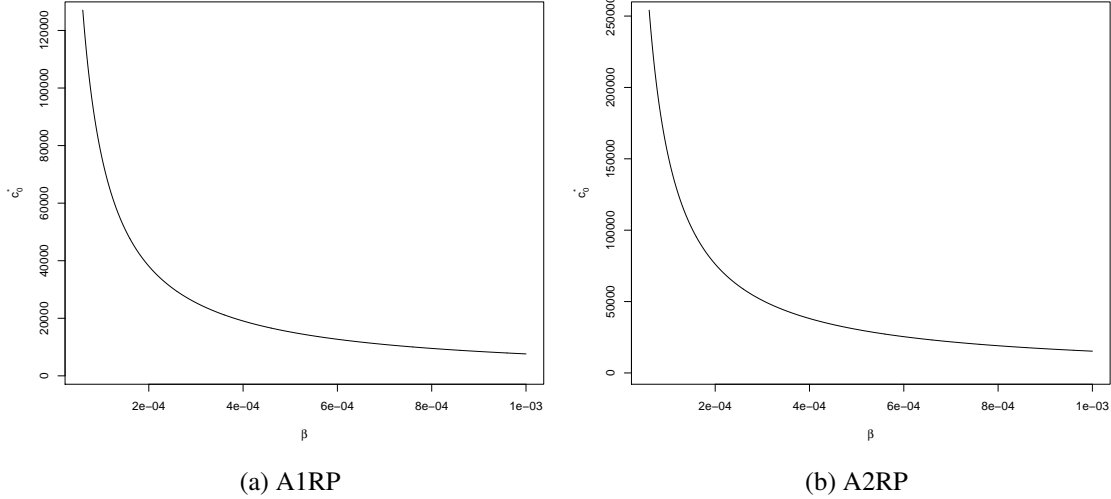


Figure 2.1: Values of  $c_0^*$  vs  $\beta$  for FSP with (a) A1RP and (b) A2RP.

## 2.6 Sequential Procedure with Stochastic Schedules

### 2.6.1 Description

In this section, we present a variant of FSP that determines the next values of the sample sizes according to the current estimates of the optimality gap and the sampling error. The larger the estimates, the larger the increases in the sample sizes. We refer to this procedure as the sequential procedure with stochastic schedules (SSP). In SSP, sample sizes when the procedure stops can be larger than FSP, but when fewer optimization problems are solved, solution times can be accelerated. Since the sample sizes are determined according to statistical estimates, they now form a stochastic schedule. We denote these stochastic schedules as  $\{M_k\}$  and  $\{N_k\}$ . In our subsequent analysis, we set  $M_k = N_k$  and  $h(n) = 1/\sqrt{n}$  and assume  $v_n^\alpha(x) = \frac{t_{n-1, \alpha} s_n(x)}{\sqrt{n}}$ . We note that  $M_k$  can be set to be a multiple of  $N_k$  and  $v_n^\alpha(x)$  is in a form satisfied by  $ArRP$ ,  $r \geq 1$ .

Similar to FSP, this variant to FSP is guided by the same objective: stopping the procedure at the first iteration when the inflated CI falls below  $\varepsilon$ . For SSP, we denote the

stopping iteration by  $K'(\varepsilon)$  and its definition is as in the FSP case

$$K'(\varepsilon) = \inf_{k \geq 1} \left\{ k : G_{N_k}(\hat{x}_k) + \frac{t_{N_k-1, \alpha} s_{N_k}(\hat{x}_k) + 1}{\sqrt{N_k}} \leq \varepsilon \right\}. \quad (2.28)$$

Suppose that at iteration  $k$  the procedure fails to stop. Then, we would like to be able to estimate the value of the sample size at the next iteration,  $N_{k+1}$ , so that the stopping criterion is satisfied. More generally, the procedure improves at the next iteration (e.g., by obtaining a better solution or by resulting in a tighter confidence interval on the optimality gap). Let us examine some of the reasons why the procedure fails to stop at iteration  $k$  in order to gain some insight into how to estimate the sample size for the next iteration. First, the current candidate solution might actually have a large optimality gap; in which case, the CI on the optimality gap at iteration  $k$  is a good or close estimate indicating not to stop yet. Second, the gap estimator might be biased. It is well-known that ArRP estimators are biased, that is, on average they tend to overestimate the optimality gap for a fixed sample size. Even though we do not exactly know the current optimality gap and the nature of bias for a particular problem, these two are approximately captured in the estimator  $G_{N_k}(\hat{x}_k)$ . Third, even when the point estimate on the optimality gap is a good or close estimate, the sampling error term of the CI might be too large. In this case, we need larger sample sizes to reduce the sampling error to stop at a fixed-width CI. Again, we do not know the exact variance, hence the sampling error, but we can use our current estimate  $s_{N_k}^2(\hat{x}_k)$  instead.

If we multiply each side of the stopping criterion in (2.28) by  $N_k$  and rearrange accordingly, then we obtain

$$-N_k \varepsilon + N_k G_{N_k}(\hat{x}_k) + \sqrt{N_k} (t_{N_k-1, \alpha} s_{N_k}(\hat{x}_k) + 1) \leq 0. \quad (2.29)$$

Replacing  $N_k$  in this inequality by a generic variable  $n$ , our goal is to find the smallest value of  $n$  such that (2.29) is satisfied. Though the estimators  $G_{N_k}(\hat{x}_k)$  and  $s_{N_k}^2(\hat{x}_k)$  in (2.29) are clearly dependent upon the sample size  $n$ , we keep these values intact in the expression as they are the best estimates we have so far for the optimality gap and the sampling error term. Therefore, we use them in calculating the next sample size. Estimating the next sample size by fixing the point estimators and solving for a value of  $n$  that satisfies

the stopping criterion is quite common in sequential statistics, such as in the two-stage, three-stage and accelerated procedures described in Ghosh et al. (1997). This approach to finding the smallest sample size necessary to attain a desired precision is analogous. The resulting sample size estimate is  $N_{k+1} = \inf_{n \in \mathbb{Z}_+} \left\{ n \geq \frac{(t_{n-1, \alpha^{SN_k}}(\hat{x}_k) + 1)^2}{(\varepsilon - G_{N_k}(\hat{x}_k))^2} \right\}$ . There are several dangers in using this sample size estimate. First, when the current optimality gap estimate is much larger than  $\varepsilon$ , which can happen easily at early iterations, the next sample size estimate  $N_{k+1}$  can actually be less than the current sample size  $N_k$ . This is rather contradictory. When the gap estimate is large, this could be an indication of either a low-quality candidate solution or high bias. In either case, a large optimality gap estimate suggests increasing the sample size in an effort to obtain a higher quality solution and/or reduce bias. On the other hand, when the gap estimate is close to  $\varepsilon$ , it results in a very large sample size increase, which is also somewhat contradictory by the same reasoning.

To remedy these deficiencies in the sample size estimation, we modify this approach and propose an alternative way to increase the sample sizes. Let  $b_k = t_{N_k-1, \alpha^{SN_k}}(\hat{x}_k) + 1$  and  $c_k = N_k G_{N_k}(\hat{x}_k)$ . Starting from (2.29), fixing  $b_k$  and  $c_k$  and replacing  $N_k$  by  $n$ , we can rewrite the stopping criterion as a quadratic expression in the square root of the sample size  $n$  as follows

$$-\varepsilon n + b_k \sqrt{n} + c_k \leq 0. \quad (2.30)$$

When the stopping rule is not satisfied at iteration  $k$ , the sample size  $N_{k+1}$  for the next iteration can be estimated as the smallest integer value for  $n$  such that the stopping rule, rewritten as in (2.30), holds. To find such a value, consider the quadratic polynomial in  $\sqrt{n}$  given in (2.30). Note that  $b_k > 0$  and  $c_k \geq 0$ . The corresponding discriminant to the quadratic polynomial can be computed as  $\Delta_k = b_k^2 + 4\varepsilon c_k \geq b_k^2 > 0$ . Hence, the quadratic polynomial in (2.30) has two roots:  $v_- = \frac{-b_k + \sqrt{\Delta_k}}{-2\varepsilon} \leq 0$  and  $v_+ = \frac{b_k + \sqrt{\Delta_k}}{2\varepsilon} > 0$ . Thus, we can select  $N_{k+1} = \lceil v_+^2 \rceil$ . This choice of  $N_{k+1}$  guarantees that  $N_{k+1} > N_k$ . Note that with  $n = N_k$ , the left-hand-side of (2.30) is positive (assuming at iteration  $k$  we did not yet stop) and  $-\varepsilon < 0$ , so, as a consequence of the properties of a quadratic polynomial with two real roots,  $N_{k+1} > N_k$ . When ArRP,  $r \geq 1$  is used, the value of  $N_{k+1}$  can be set to the smallest integer multiple of  $r$  that is greater than or equal to  $\lceil v_+^2 \rceil$  (so that  $N_{k+1}$  is divisible by  $r$ ).

This is a somewhat heuristic way to increase the sample sizes but it has the right implication. For instance, all else equal, if the optimality gap estimate is larger, then the sample size increases will be larger. Similarly, if the variance estimate is larger, the sample size increases will be larger as well. We note that this might not be the only way to estimate the next sample sizes. For instance, more elaborate adaptive sequential procedures can be designed aiming to correct bias and/or variance but we do not pursue it here and leave it for future work. Nevertheless, with these sample size increases we are able to show results similar to FSP's but under more restrictive conditions. Below, we first provide a brief statement of the sequential procedure with stochastic sample size schedules (SSP). Then, we prove and discuss several theoretical properties.

**SSP:**

**Input:** Initial sample sizes  $m_0 = n_0$ , values of  $\varepsilon > 0$ ,  $0 < \alpha < 1$ , and resampling frequencies  $k_f^n$  and  $k_f^m$ .

**Output:** A candidate solution  $\hat{x}_{K'(\varepsilon)}$ , and a confidence interval on its optimality gap with a width of  $\varepsilon$ .

**Step 0.** (Initialization) Set  $k = 1$ ,  $M_1 = N_1 = n_0$ , generate  $M_1$  observations  $\xi_1^1, \xi_1^2, \dots, \xi_1^{M_1}$  and independently generate  $N_1$  observations  $\xi_2^1, \xi_2^2, \dots, \xi_2^{N_1}$ .

**Step 1.** (Generate candidate solution) Using the observations  $\xi_1^1, \dots, \xi_1^{M_k}$ , solve  $(SP_{M_k})$  to obtain  $x_{M_k}^*$ . Set  $\hat{x}_k = x_{M_k}^*$ .

**Step 2.** (Assess solution quality) Using the observations  $\xi_2^1, \dots, \xi_2^{N_k}$ , calculate  $G_{N_k}(\hat{x}_k)$ ,  $s_{N_k}^2(\hat{x}_k)$  and  $v_{N_k}^\alpha(\hat{x}_k)$ .

**Step 3.** (Check stopping criterion) If  $G_{N_k}(\hat{x}_k) + v_{N_k}^\alpha(\hat{x}_k) + 1/\sqrt{N_k} \leq \varepsilon$ , then set  $K'(\varepsilon) = k$ ,  $N'(\varepsilon) = N_k$ ,  $M'(\varepsilon) = M_k$ , and  $\hat{x}_{K'(\varepsilon)} = \hat{x}_k$  and stop. Output candidate solution  $\hat{x}_{K'(\varepsilon)}$  with confidence interval on  $\mu_{\hat{x}_{K'(\varepsilon)}}$  as  $[0, \varepsilon]$ .

**Step 4.** (Improve: Increase sample size)

**4.1.** Solve the following quadratic equation in terms of  $\sqrt{n}$ :

$$\frac{1}{n} (N_k G_{N_k}(\hat{x}_k)) + \frac{1}{\sqrt{n}} (t_{N_k-1, \alpha} s_{N_k}(\hat{x}_k) + 1) = \varepsilon.$$

- 4.2.** The above quadratic equation has two roots,  $v_+ > 0$  and  $v_- \leq 0$ . Thus, there is a unique solution of  $\sqrt{n} = v_+$ . Consequently,  $N_{k+1}$  is chosen as  $N_{k+1} = \lceil v_+^2 \rceil$ .
- 4.3.** Set  $M_{k+1} = N_{k+1}$ . If  $k_f^m$  divides  $k + 1$  then sample  $\xi_1^1, \xi_1^2, \dots, \xi_1^{M_{k+1}}$  independently of samples generated in previous iterations. Else, sample  $M_{k+1} - M_k$  additional observations,  $\xi_1^{M_k+1}, \dots, \xi_1^{M_{k+1}}$ . Similarly, if  $k_f^n$  divides  $k + 1$  then sample  $\xi_2^1, \xi_2^2, \dots, \xi_2^{N_{k+1}}$  independently of samples generated in previous iterations. Else, sample  $N_{k+1} - N_k$  additional observations,  $\xi_2^{N_k+1}, \dots, \xi_2^{N_{k+1}}$ . Set  $k = k + 1$  and go to 1.

Like FSP, the sample sizes when the procedure stops are denoted by  $N'(\varepsilon) = N_{K'(\varepsilon)}$  and  $M'(\varepsilon) = m_{K'(\varepsilon)}$ . In SSP, sample sizes when the procedure stops can be larger than FSP, but when fewer optimization problems are solved, solution times can be accelerated. The overall solution time, of course, depends on the size of the optimization problems solved as well as the number. In early iterations, the statistical estimators may be poor and/or the candidate solution may be of low-quality, leading to large values of the next sample sizes. In this case, larger optimization problems need to be solved and benefits from numerical speedup techniques such as warm-starting may be lost. The deterministic schedules in FSP can be increased slowly, such as of order  $O(\ln^2 k)$ . This can be desirable if, for instance, obtaining samples is costly, the solution procedure allows for quick updates (e.g., via warm-starting), or solving a problem with a large sample size is computationally burdensome. In contrast, in SSP, the sample sizes are increased in random jumps given the current gap and variance estimates. The larger the estimates, the larger the increases in sample sizes.

### 2.6.2 Asymptotic Validity, Finite Stopping, and Finite Sample Sizes

We now provide equivalent results for SSP as the ones stated in Theorem 2 for FSP. Namely, we present conditions under which SSP finds an  $\varepsilon$ -optimal solution with probability one as  $\varepsilon \downarrow 0$  and for  $\varepsilon > 0$  it stops in a finite number of iterations using finite number of samples, w.p.1.

**Theorem 3.** Assume (A4) holds.

(i) Consider SSP that stops at iteration  $K'(\varepsilon)$  according to stopping criterion (2.28), defined in terms of  $\varepsilon$ , using  $N'(\varepsilon) = N_{K'(\varepsilon)}$  and  $M'(\varepsilon) = m_{K'(\varepsilon)}$  samples. Then,

$$\lim_{\varepsilon \downarrow 0} \mathbb{P} \left( \mu_{\hat{x}_{K'(\varepsilon)}} \leq \varepsilon \right) = 1.$$

(ii) In addition, assume (SP2) has a unique optimum solution,  $X^* = \{x^*\}$  and that SSP uses ArRP,  $r \geq 1$ , based on SRP given in (2.3)-(2.4), to calculate the confidence intervals.

Given  $\varepsilon > 0$ ,

$$\mathbb{P}(K'(\varepsilon) < \infty) = \mathbb{P}(N'(\varepsilon) < \infty) = \mathbb{P}(M'(\varepsilon) < \infty) = 1.$$

*Proof.* (i) Let  $Y_m = \mu_{x_m^*} = \mathbb{E}[f(x_m^*, \tilde{\xi}) | x_m^*] - z^*$  for some  $m \rightarrow \infty$ . By (A4),  $Y_m = 0$ , w.p.1 for  $m$  large enough. Similarly define  $Y_{M'(\varepsilon)} = \mu_{x_{M'(\varepsilon)}^*} = \mathbb{E}[f(x_{M'(\varepsilon)}^*, \tilde{\xi}) | x_{M'(\varepsilon)}^*] - z^*$  and note that  $\hat{x}_{K'(\varepsilon)} = x_{M'(\varepsilon)}^*$ . For SSP,  $M'(\varepsilon) \rightarrow \infty$ , w.p.1 as  $\varepsilon \downarrow 0$ . Therefore, using Proposition 1, we obtain that  $Y_{M'(\varepsilon)} = 0$ , w.p.1 for  $\varepsilon$  small enough. The desired result follows.

(ii) Let  $G_{N_k} = G_{N_k}(\hat{x}_k)$ ,  $s_{N_k} = s_{N_k}(\hat{x}_k)$ ,  $t_k = t_{N_k-1, \alpha}$  and  $\Delta_k = (t_k s_{N_k} + 1)^2 + 4\varepsilon N_k G_{N_k}$ . Note that

$$\mathbb{P}(K'(\varepsilon) = \infty) \leq \mathbb{P} \left( \bigcap_{k=1}^{\infty} \{N_{k+1} > N_k\} \right) \quad (2.31)$$

$$= \mathbb{P} \left( \bigcap_{k=1}^{\infty} \left\{ \left[ \left( \frac{(t_k s_{N_k} + 1) + \sqrt{\Delta_k}}{2\varepsilon} \right)^2 \right] > N_k \right\} \right). \quad (2.32)$$

The right-hand side of (2.32) is positive only if  $N_k G_{N_k} \rightarrow \infty$  and  $s_{N_k} \rightarrow \infty$  with positive probability as  $k \rightarrow \infty$ . Let  $A = \{\omega : K'(\varepsilon) = \infty\}$ . Suppose  $\mathbb{P}(A) > 0$ . Conditioned on  $A$ ,  $M_k = N_k \rightarrow \infty$ , w.p.1 as  $k \rightarrow \infty$ . For every  $k = 1, 2, 3, \dots$  construct a decreasing sequence of  $\bar{\varepsilon}$  such that  $k = \lceil \frac{1}{\bar{\varepsilon}} \rceil$ . Set  $M_k = M(\bar{\varepsilon})$  and  $N_k = N(\bar{\varepsilon})$ . Then, conditioned on  $A$ ,  $M(\bar{\varepsilon}) = N(\bar{\varepsilon}) \rightarrow \infty$ , w.p.1 as  $\bar{\varepsilon} \downarrow 0$ . Therefore, by Lemma 3, conditioned on  $A$ ,  $N_k G_{N_k} \rightarrow 0$  and  $s_{N_k} \rightarrow 0$ , w.p.1 as  $k \rightarrow \infty$ . This implies that right-hand side of (2.32) is 0, which is a contradiction. This shows  $\mathbb{P}(K'(\varepsilon) < \infty) = 1$ . Now,  $\mathbb{P}(N'(\varepsilon) = \infty)$  is also less than or equal to the right-hand side of (2.31), hence is 0. By noting that  $M'(\varepsilon) = N'(\varepsilon)$ , we have  $\mathbb{P}(M'(\varepsilon) < \infty) = 1$ .  $\square$

Conditions in part (ii) of Theorem 3 are more restrictive than those of Theorem 2. In order to ensure finiteness of the number of iterations and the sample sizes used when SSP stops, we assumed unique optimality and focused on using ArRP for assessing solution quality. Under these conditions, we can further show additional equivalent results for SSP like those of FSP. We do this in the next two propositions. First, we show that for  $\varepsilon > 0$ , SSP stops in a finite expected number of iterations, i.e.,  $\mathbb{E}[K'(\varepsilon)] < \infty$ . Then, by a slight modification in the initial sample sizes, we obtain asymptotic sample size increases for SSP similar to FSP. That is, asymptotically  $N(\varepsilon) = M(\varepsilon)$  grows of order  $O(\varepsilon^{-2})$ , w.p.1.

**Proposition 5.** *Assume (A5) holds and (SP2) has a unique optimal solution,  $X^* = \{x^*\}$ . Consider SSP that uses ArRP,  $r \geq 1$ , based on SRP given in (2.3)-(2.4), to calculate the confidence intervals and stops at iteration  $K'(\varepsilon)$  according to stopping criterion (2.28), defined in terms of  $\varepsilon$ , using  $N'(\varepsilon) = N_{K'(\varepsilon)}$  and  $M'(\varepsilon) = m_{K'(\varepsilon)}$  samples. Given  $\varepsilon > 0$ ,  $\mathbb{E}[K'(\varepsilon)] < \infty$ .*

*Proof.* Let  $B_{N_k}$  denote the event  $\left\{G_{N_k}(\hat{x}_k) + v_{N_k}^\alpha(\hat{x}_k) + h(N_k) > \varepsilon\right\}$ . Then,

$$\begin{aligned} \mathbb{E}[K'(\varepsilon)] &= 1 + \sum_{k=1}^{\infty} P(K'(\varepsilon) > k) \\ &\leq 1 + \mathbb{P}(B_{N_1}) + \sum_{k=2}^{\infty} \mathbb{P}(B_{N_k}). \end{aligned} \quad (2.33)$$

Note that  $N_k$  is an integer-valued random variable that takes on values  $N_k \geq n_0 + k - 1$  for  $k \geq 2$  and  $\hat{x}_k$  is obtained by solving an independent sampling problem with sample size  $M_k = N_k$ . Using arguments similar to (2.16) and continuing from (2.33),

$$\begin{aligned} \mathbb{E}[K'(\varepsilon)] &\leq 2 + \sum_{k=2}^{\infty} \sum_{N_k=n_0+k-1}^{\infty} \mathbb{P}(B_{N_k}) \\ &\leq 2 + C(r+1)e^{-\beta/r(n_0-1)} \sum_{k=2}^{\infty} \sum_{j=0}^{\infty} e^{-\beta/r(k+j)}, \end{aligned} \quad (2.34)$$

where (2.34) follows from (A5). The right hand side of (2.34) is finite for  $\beta > 0$ .  $\square$

We now modify SSP so that the initial sample sizes are selected according to

$$m_0 = n_0 = \left\lceil \max \left\{ \bar{n}_0, \ln \frac{1}{\varepsilon} \right\} \right\rceil, \quad (2.35)$$

where  $\bar{n}_0 \geq 2$  is a finite constant (for instance, one can set  $\bar{n}_0 = 50$ ). The initial sample sizes (2.35) are equal to  $\bar{n}_0$  for all practical purposes but tend to infinity as  $\varepsilon \downarrow 0$ . For example, with  $\bar{n}_0 = 50$ , values of  $\varepsilon < e^{-50}$  will result in  $n_0 > 50$ . By selecting the sample sizes as in (2.35), one is willing to use more samples as more precision is demanded. With the initial sample sizes defined as above, it is possible to show that the sample sizes used by SSP,  $N'(\varepsilon) = M'(\varepsilon)$ , asymptotically increase at the same rate as  $1/\varepsilon^2$ , w.p.1.

**Proposition 6.** *Suppose (A4) holds and (SP2) has a unique optimal solution,  $X^* = \{x^*\}$ . Let  $m_0 = n_0$  be chosen according to (2.35). Consider SSP that uses ArRP,  $r \geq 1$ , based on SRP given in (2.3)-(2.4), to calculate the confidence intervals and stops at iteration  $K'(\varepsilon)$  according to stopping criterion (2.28), defined in terms of  $\varepsilon$ , using  $N'(\varepsilon) = N_{K'(\varepsilon)}$  and  $M'(\varepsilon) = m_{K'(\varepsilon)}$  samples. Then,*

$$\lim_{\varepsilon \downarrow 0} \frac{M'(\varepsilon)}{1/\varepsilon^2} = 1, \quad \text{and} \quad \lim_{\varepsilon \downarrow 0} \frac{N'(\varepsilon)}{1/\varepsilon^2} = 1, \quad \text{w.p.1.}$$

*Proof.* Let  $G_{N_k} = G_{N_k}(\hat{x}_k)$ ,  $s_{N_k} = s_{N_k}(\hat{x}_k)$ ,  $t_k = t_{N_k-1, \alpha}$  and let  $\Delta_k = (t_k s_{N_k} + 1)^2 + 4\varepsilon N_k G_{N_k}$ . When SSP stops, (2.28) is satisfied. Rearranging (2.28), we have

$$\frac{\left( \sqrt{N'(\varepsilon)} G_{N'(\varepsilon)} + t_{K'(\varepsilon)} s_{N'(\varepsilon)} + 1 \right)^2}{\varepsilon^2} \leq N'(\varepsilon). \quad (2.36)$$

Based on how the sample sizes are increased, we also have

$$N'(\varepsilon) \leq n_0 + r + \frac{\left( t_{K'(\varepsilon)-1} s_{N_{K'(\varepsilon)-1}} + 1 + \sqrt{\Delta_{K'(\varepsilon)-1}} \right)^2}{4\varepsilon^2}, \quad (2.37)$$

where we set  $M_{K'(\varepsilon)-1} = N_{K'(\varepsilon)-1} = n_0$  when  $K'(\varepsilon) = 1$ . Dividing each term in (2.36) and (2.37) by  $1/\varepsilon^2$ , taking limits as  $\varepsilon \downarrow 0$  and following the same line of logic as in the proof of Proposition 4, we obtain

$$1 \leq \lim_{\varepsilon \downarrow 0} \frac{N'(\varepsilon)}{1/\varepsilon^2} \leq 1, \quad \text{w.p.1.} \quad (2.38)$$

As  $N'(\varepsilon) = M'(\varepsilon)$ , the same result holds for  $M'(\varepsilon)$ .  $\square$

The technical reason for selecting  $n_0$  according to (2.35) becomes apparent in the proof of Proposition 6. This hinges on the fact that unlike FSP, as  $\varepsilon \downarrow 0$ , even though

$M'(\varepsilon) = N'(\varepsilon) \rightarrow \infty$ , w.p.1, the same is not true of  $K'(\varepsilon)$ . With  $m_0 = n_0$  fixed (not depending on  $\varepsilon$ ), as  $\varepsilon \downarrow 0$ ,  $K'(\varepsilon)$  can be 2 (or any other arbitrary constant) with positive probability. In this case,  $G_{N_{K'(\varepsilon)-1}}(\hat{x}_{K'(\varepsilon)-1}) = G_{n_0}(x_{m_0}^*)$  and  $s_{N_{K'(\varepsilon)-1}}^2(\hat{x}_{K'(\varepsilon)-1}) = s_{n_0}^2(x_{m_0}^*)$  are random variables that do not depend on  $\varepsilon$ . Hence, the right-hand limit in (2.38) cannot be guaranteed to be 1, w.p.1. However, selecting the sample sizes according to (2.35) ensures that  $n_0$  tends to infinity as  $\varepsilon \downarrow 0$ .

## 2.7 Summary and Concluding Remarks

In this chapter, we developed sequential stopping rules for Monte Carlo sampling-based procedures for a class of stochastic programs (SP2). Two methods were presented: the fully sequential procedure (FSP) and the sequential procedure with stochastic schedules (SSP). In FSP, we solve a sequence of sampling approximation problems, with an increasing deterministic sample size schedule, to generate candidate solutions. The quality of the candidate solutions is determined by building a confidence interval on their optimality gap. At least a second sampling approximation problem is solved in order to calculate the point estimate and sampling error on the optimality gap. FSP stops when the width of the CI on the optimality gap plus an inflation factor falls below a pre-determined tolerance,  $\varepsilon$ . SSP is similar to FSP, except that the sample sizes for the next iteration are determined based on current statistical estimates of the optimality gap and sampling error term. Key theoretical results, such as finite stopping and asymptotic validity of the quality of the solution found, were established for each procedure. In the next chapter, we examine the performance of these procedures on test problems from the literature.

## CHAPTER 3

## FWSSM: COMPUTATIONAL RESULTS

In this chapter, we empirically examine the small sample behavior of the sequential sampling procedures developed in Chapter 2. We use two test problems, APL1P and 10D, whose solutions are known, to study the full extent of the performance of our procedures and to examine different parameter settings. APL1P is an instance of the electric power generation model described in Example 1.2 with 5 independent stochastic parameters and 1280 scenarios (Infanger, 1992); and 10D is an instance of the stochastic knapsack model described in Example 1.1 (Kleywegt et al., 2001) with 10 decision variables (10 objects to select from) and 10 independent normally distributed stochastic parameters (the weights of each of the objects). We choose these two problems to examine our procedures for two reasons. First, both problems are known to be challenging for solution via a sampling approximation. That is, the sampling approximations to these problems are known to have a low rate of convergence, i.e.,  $\beta$  in (A5) is small for these problems. Second, since APL1P has a small number of realizations, and 10D has normally distributed random variables, we can calculate the actual optimality gaps to verify our procedures. Both problems satisfy assumptions (A4) and (A5) under i.i.d. sampling and have unique optimal solutions. Then, in order to illustrate the application of the procedures in practice (i.e., when  $z^*$  and true optimality gaps are not known a priori), we experiment on an instance of the motor freight scheduling model described in Example 1.3 called 20TERM. This test problem, 20TERM, corresponds to a large scale two-stage stochastic linear program with 40 stochastic parameters and  $1.1 \times 10^{12}$  scenarios (Mak et al., 1999). Due to its large scale, its optimal solution is not known. Note that this problem also satisfies assumptions (A4) and (A5) under i.i.d. sampling.

This chapter is organized as follows. In §3.1, we investigate the characteristics of the two test problems APL1P and 10D with respect to sampling approximation problems. Then, in §3.2, we analyze the effect of resampling to determine the best set of values of  $k_f^m$

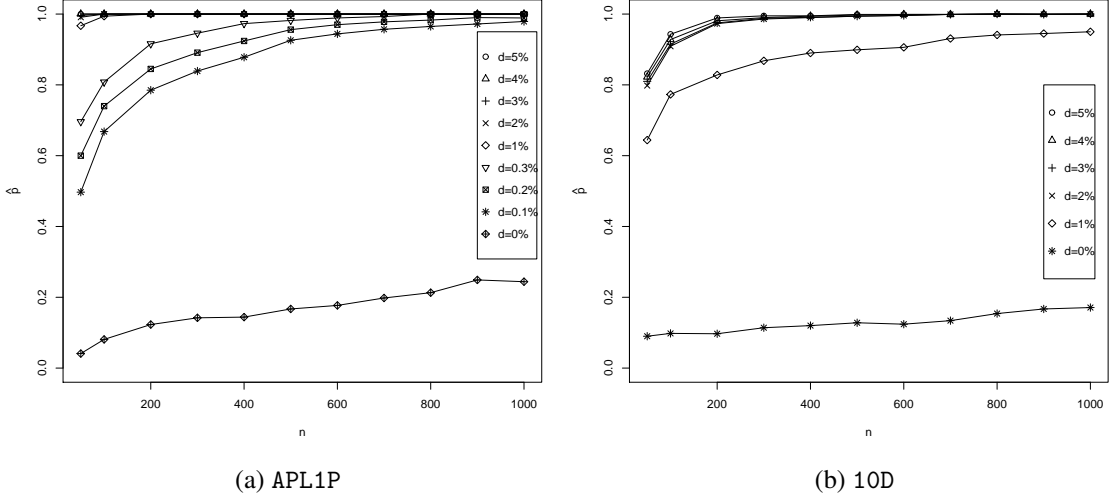


Figure 3.1: Probability of obtaining  $\varepsilon$ -optimal solutions for (a) APL1P and (b) 10D.

and  $k_f^n$  for the implementation of the procedures. In §3.3, we apply FSP and SSP to APL1P and 10D and compare the empirical performances of the two procedures. We examine the behavior of the procedures as  $\varepsilon \downarrow 0$  on these two test problems in §3.4. Next, in §3.5, we report results on 20TERM, providing guidelines on how to choose  $\varepsilon$  and illustrating how to implement the procedures on a general model whose solution is unknown. Finally, we close this chapter by providing a summary of our findings and presenting implementation guidelines in §3.6.

### 3.1 Test Problems: APL1P and 10D

Before we present results of our computational experiments with FSP and SSP, we first examine the two test problems APL1P and 10D in more detail to gain insight. Figure 3.1 shows estimates of probability of obtaining  $\varepsilon$ -optimal solutions for APL1P and 10D. That is, it shows estimated values of  $P(\mu_{x_n^*} \leq \varepsilon)$  for various values of  $n$ . To obtain these estimates we solved 1,000 independent sampling problems ( $SP_n$ ) for each value of  $n$  and calculated the frequency of the  $\varepsilon$ -optimal solutions obtained out of these 1,000 sampling problems. We set  $\varepsilon$  to different percentages of the optimal objective function value,  $\varepsilon =$

$d \cdot z^*$ , ranging  $d$  from  $d = 0\%$  to  $d = 5\%$ . The estimated probabilities in Figure 3.1 have 90% confidence interval half-widths of at most 0.01. It appears that for both problems, the probability of obtaining optimal solutions is indeed quite low, even for large sample sizes. However, as  $\varepsilon$  increases, the probability of obtaining  $\varepsilon$ -optimal solutions dramatically increases. This indicates that the fixed-width stopping rules could perform well when used with moderate values of  $\varepsilon$ .

### 3.2 Effect of Resampling

In this section, we examine the effect of augmenting the set of observations with a few additional ones versus getting rid of all of previously used observations and generating an entire set of new observations through resampling. Recall that both FSP and SSP include a resampling scheme controlled by the parameters  $k_f^m$  and  $k_f^n$  for the set of observations used for generating candidate solutions and for assessing solution quality, respectively. When sampled observations are augmented, estimates in the next iteration are dependent on current estimates. In this case, when warm-starting is used, solution time can be reduced. When observations are resampled from time to time, then we can avoid getting stuck in a bad sample path and improve both on solution time and the coverage probability. It has been observed for another stopping rule that as the resampling frequency is increased (i.e.,  $k_f^n$  or  $k_f^m$  is decreased), the sequential sampling procedure stops on average in fewer number of iterations. However, this can lower the coverage probability below desired levels (Bayraksan and Morton, 2011). Our goal in this section is to find the values of  $k_f^m$  and  $k_f^n$  that result in minimal solution time and stopping iteration, and consequently sample sizes used, without sacrificing the targeted coverage probability.

To gain an understanding of the effect of resampling, we applied FSP using AIRP with sample size schedules  $m_k = n_k = n_0 + 2(k - 1)$  with  $n_0 = 50$  on both test problems APL1P and 10D. We set  $\varepsilon = 0.2\%z^*$  for APL1P and  $\varepsilon = 2\%z^*$  for 10D. The other parameters are set to  $h(n) = 1/\sqrt{n}$ , and  $\alpha = 0.10$ . We then tested three resampling schemes: (i) we set  $k_f^m = \infty$  (always augment observations used for assessing solution quality) and only let  $k_f^n$  vary, (ii) we set  $k_f^n = \infty$  (always augment observations used for generating candidate

solutions) and only let  $k_f^m$  vary, and (iii) we set both resampling frequencies to be the same,  $k_f^n = k_f^m$  and let them vary. In each of the three settings, we used resampling frequencies of 1 (resample at every iteration), 3, 6, 12, 25, 50, 75, and 100 (resample independent set of observations every 100 iterations). We ran the procedures for each configuration 100 times and report the averages. We also conducted these experiments using A2RP and found similar results. For brevity, we only report our findings for A1RP.

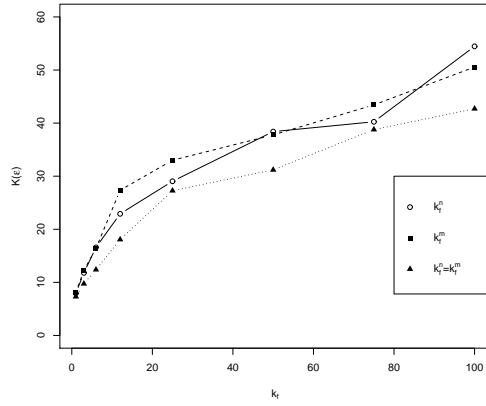
Figure 3.2 shows the results of our experiments. We denote scheme (i) as  $k_f^n$ , scheme (ii) as  $k_f^m$  and scheme (iii) as  $k_f^n = k_f^m$ . The first column of Figure 3.2 corresponds to APL1P and the second column to 10D. From top to bottom, the first row shows changes in the stopping iteration  $K(\varepsilon)$ , the second row the solution time of the procedure (in seconds) and the third row shows changes in coverage probability,  $P(\mu_{\hat{x}_{K(\varepsilon)}} \leq \varepsilon)$ , as  $k_f^n$  and  $k_f^m$  vary. In all three resampling schemes, the more often we resample, the sooner the procedure stops. Performances of the three schemes with respect to  $K(\varepsilon)$  and running time are closest when values of  $k_f^n$ ,  $k_f^m$  are small ( $\leq 6$ ). As  $k_f^n$ ,  $k_f^m$  increase, differences in this respect become apparent. In particular,  $k_f^n = k_f^m$  scheme tends to stop earlier compared to the other two. This behavior is observed more consistently for APL1P than for 10D. In terms of coverage probability, however, this does not seem to have an adverse effect.

For APL1P, significant differences exist across schemes as depicted in Figure 3.2 (e). Schemes  $k_f^m$  and  $k_f^n = k_f^m$  result in similar coverage probabilities whereas coverage probability of  $k_f^n$  is consistently lower. We note that the coverage probabilities of the three schemes converge to the same value as  $k_f^n$ ,  $k_f^m$  increase. The differences in Figure 3.2 (e) can be explained as follows. If the procedure starts with a *bad solution* (a solution that has an optimality gap higher than  $\varepsilon$ ), when no resampling is employed and the current sample path for assessing solution quality is a *good* sample path (i.e., that sample path results in CI's with width larger than  $\varepsilon$  for *bad solutions* and with width smaller than  $\varepsilon$  for *good solutions*), the procedure will tend to run longer until a *good solution* (an  $\varepsilon$ -optimal solution) is obtained. There is, of course, some noise in this process, i.e., there can be instances where the procedure might stop at a bad solution. When resampling scheme  $k_f^m$  is introduced, this allows the process to jump from a sample that generates bad solutions to one that generates good solutions. This way, the procedure can stop earlier at a

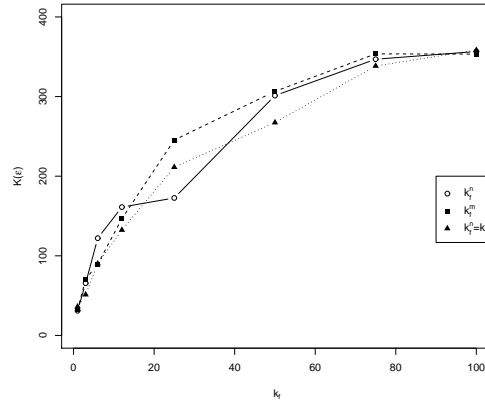
good solution. In contrast, under resampling scheme  $k_f^n$ , only the sample used to assess candidate solutions is regenerated. So, when the procedure starts with a bad solution, disrupting assessing solution quality causes early stopping with a bad solution.

To verify our hypothesis and gain more insight, we examined the three schemes in more detail. In Table 3.1, we provide a breakdown of the quality of candidate solutions for all three resampling schemes at  $k_f^n = 1$ ,  $k_f^m = 1$  and  $k_f^m = k_f^n = 1$ . We denote  $\varepsilon$ -optimal solutions as  $G$  (good) and all other solutions as  $B$  (bad). In Table 3.1, the columns provide a breakdown of the quality of candidate solutions at the first iteration and the rows provide a breakdown of the quality of the candidate solutions when the procedure stops for the three resampling schemes. Of the 100 runs made, 56 start with a good solution for all three configurations  $k_f^m$ ,  $k_f^n$ , and  $k_f^m = k_f^n$ . The solution breakdown at the start is the same for all the configurations since the same random number streams are used in all three cases. When the procedure stops, however,  $k_f^m$  and  $k_f^m = k_f^n$  end up with 76 and 75 good solutions respectively, while  $k_f^n$  finished with only 61 good solutions. Let us focus on the case where the procedure starts with a bad solution (columns  $B$ ). In this case, resampling the observations that generate the candidate solutions acts as a corrective action that drives the procedure away from bad solutions and towards good solutions (for the most part). Such is the case for  $k_f^m$  where 35 of the 44 runs that started with a bad solution ended up with a good solution. For  $k_f^n$ , however, only 6 of the 44 runs starting with bad solutions ended up with good solutions. Results for  $k_f^m = k_f^n$  are similar to  $k_f^m$  where 30 of 44 runs that started with a bad solution ended up with a good solution. When the procedure starts with a good solution (columns  $G$ ),  $k_f^n$  scheme once again adjusts itself to the candidate solution and 55 out of 56 runs starting with good solutions stopped at a good solution. Resampling scheme  $k_f^m$ , on the other hand, has an adverse effect, 15 out of 56 runs starting with good solutions ended up at bad solutions. This adverse effect is, however, not significant enough to negatively affect coverage probability and is mitigated by the substantial gain in good solutions made from the runs starting with bad solutions. Scheme  $k_f^m = k_f^n$  in this case is somewhere in between, and closer to  $k_f^m$ , with 45 out of the 56 runs that started with good solutions ending with good solutions.

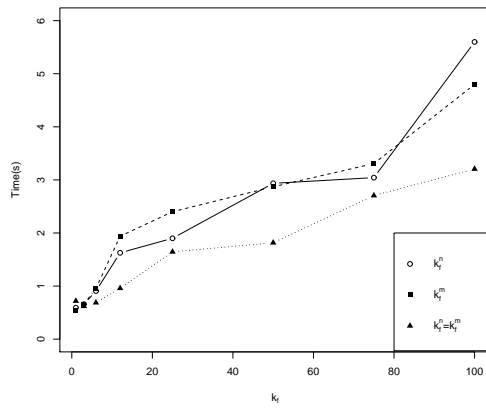
For 10D, the coverage probabilities remain relatively close for all three resampling



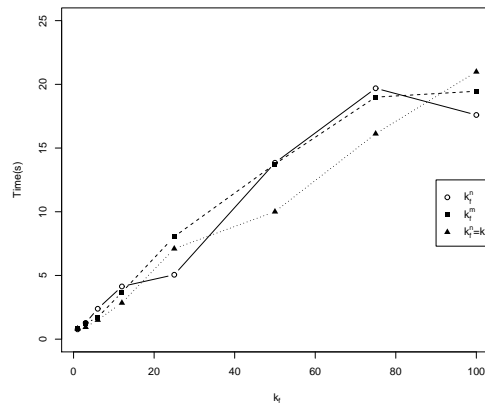
(a) APL1P:  $K(\epsilon)$



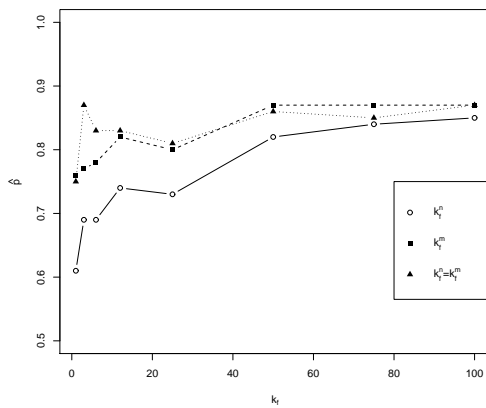
(b) 10D:  $K(\epsilon)$



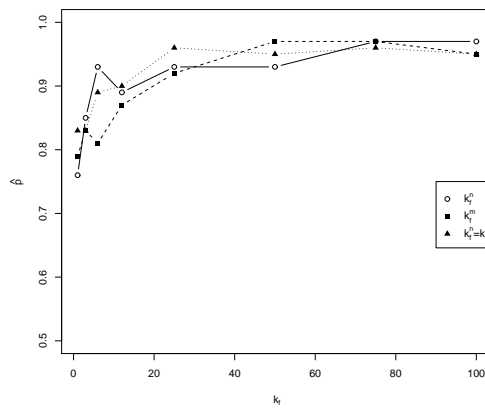
(c) APL1P: Time



(d) 10D: Time



(e) APL1P:  $\hat{p}$



(f) 10D:  $\hat{p}$

Figure 3.2: Effect of resampling.

	<i>G</i>	<i>B</i>	<i>Total</i>		<i>G</i>	<i>B</i>	<i>Total</i>		<i>G</i>	<i>B</i>	<i>Total</i>
<i>G</i>	55	6	61	<i>G</i>	41	35	76	<i>G</i>	45	30	75
<i>B</i>	1	38	39	<i>B</i>	15	9	24	<i>B</i>	11	14	25
<i>Total</i>	56	44	100	<i>Total</i>	56	44	100	<i>Total</i>	56	44	100
	(a) $k_f^n$				(b) $k_f^m$				(c) $k_f^m = k_f^n$		

Table 3.1: Solutions quality breakdown for APL1P.

schemes except when the value of the resampling frequency is 6. When resampling is performed every 6 iterations, the resampling scheme  $k_f^m$  records much lower coverage probability results than the other two resampling schemes. This difference can be ascribable to the noise in the process due to the effect of randomness.

As mentioned before, we also conducted a similar analysis using A2RP. The results are similar, except the coverage probabilities (for APL1P) are higher for A2RP (see Figure 3.3). This agrees with the behavior of SRP and A2RP in non-sequential settings (Bayraksan and Morton, 2006). Our criterion for choosing a particular set of values for the resampling frequencies is to minimize computational effort ( $K(\epsilon)$ , solution time) while obtaining a coverage probability of at least  $1 - \alpha$ . From the analysis we have done, setting  $k_f^m = k_f^n = 3$  satisfies this requirement. Therefore, we will use this set of values of the resampling frequencies for the computational experiments upon which we report in the next sections.

### 3.3 Results on APL1P and 10D

In this section, we study the empirical performance of the fully sequential procedure and the procedure with stochastic schedules on the test problems APL1P and 10D. We applied FSP using SRP for the CI on the optimality gap with sample size schedules  $n_k = m_k = n_0 + 2(k - 1)$ , and using A2RP for the CI on the optimality gap with sample size schedules  $n_k = m_k = n_0 + 2(k - 1)$  and  $n_k = m_k = n_0 + 100(k - 1)$  for values of the initial sample size ranging from  $n_0 = 50$  to  $n_0 = 1000$ . We applied SSP using A2RP for the CI on the optimality gap. Other parameter settings are as described in §3.2. In the subsequent graphs and tables, the fully sequential procedure with SRP and A2RP is

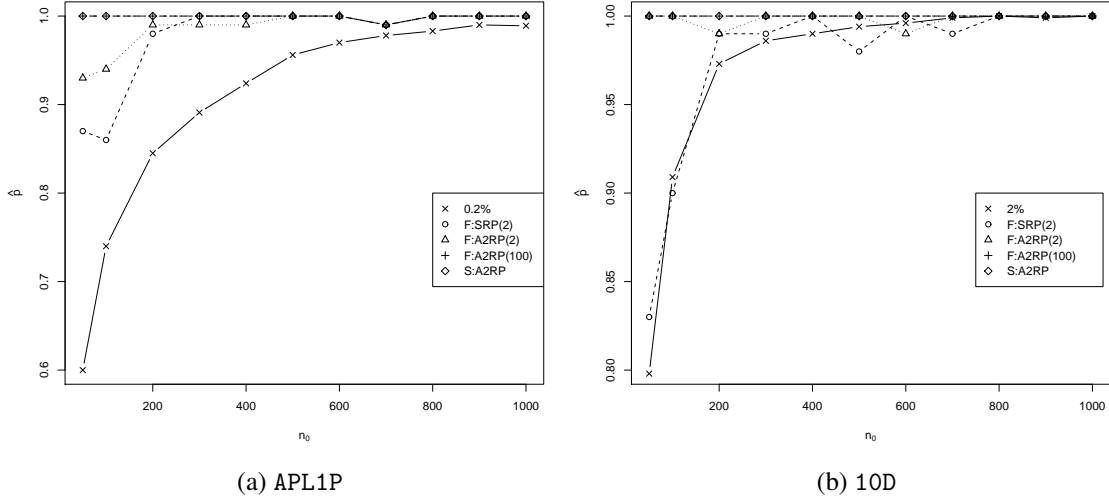


Figure 3.3: Coverage probabilities for (a) APL1P and (b) 10D.

denoted by F:SRP(2), F:A2RP(2), and F:A2RP(100), where the number in parentheses distinguishes between the cases where the sample size is increased two by two and one hundred by one hundred from one iteration to the next. The procedure with stochastic schedules using A2RP is denoted by S:A2RP. We ran each procedure 100 times for each initial sample size value  $n_0$  and report 90% confidence intervals on the means of the different performance measures (coverage probability, sample sizes used, stopping iteration, and solution time).

Figure 3.3 shows the empirical coverage probabilities obtained as a result of the computational experiments for APL1P and 10D. The “0.2%” line in Figure 3.3a and the “2%” line in Figure 3.3b show the probability of obtaining  $\epsilon$ -optimal solutions for APL1P and 10D respectively as the sample size  $n_0$  varies. These are the values from Figure 3.1. All procedures seem to have good coverage probabilities, except for F:SRP(2) for small initial sample sizes (specifically for  $n_0 = 50$  and  $n_0 = 100$ ). SSP consistently finds  $\epsilon$ -optimal solutions with high probability.

Tables 3.2 and 3.3 list the sample sizes used when the procedures terminate along with the stopping iteration and the solution time (in seconds) for APL1P and 10D, respectively. All values are listed along with 90% confidence interval half-widths around their esti-

mated means. The fully sequential procedure with SRP uses fewer sample sizes than all the other procedures while the procedure with stochastic schedules uses the most. There is a considerable difference between the sample sizes used by SSP and FSP, especially at small initial sample sizes,  $n_0$ . For APL1P, as the initial sample size increases, this difference decreases. This is because at higher sample sizes, the procedure with stochastic schedules stops quicker (see results for  $K(\varepsilon)$  and  $K'(\varepsilon)$  in Table 3.2), and also the sample size increases are not that large since the estimators are more reliable at larger initial sample sizes. For 10D, the estimators have a large bias; even at larger initial sample sizes bias remains. This results in larger jumps in SSP and the sample sizes used by the procedure with stochastic schedules remain larger compared to the fully sequential procedures at all levels of initial sample sizes,  $n_0$ .

Performance Measure	$n_0$	F:SRP(2)	F:A2RP(2)	F:A2RP(100)	S:A2RP
Sample sizes used:	50	67.48 ± 2.58	106.26 ± 6.90	409.00 ± 34.37	3704.68 ± 1018.42
	100	111.26 ± 2.19	138.76 ± 5.64	383.00 ± 34.21	1904.78 ± 521.12
	200	206.22 ± 1.56	214.90 ± 2.65	459.00 ± 35.70	1434.06 ± 264.63
	300	304.44 ± 1.21	310.00 ± 2.05	478.00 ± 32.19	1242.84 ± 193.17
	400	402.48 ± 0.63	406.20 ± 1.41	535.00 ± 26.59	1108.10 ± 187.94
	500	501.44 ± 0.59	504.32 ± 1.08	594.00 ± 19.95	980.42 ± 127.33
	600	601.12 ± 0.42	602.76 ± 0.82	693.00 ± 23.64	1031.58 ± 118.07
	700	700.58 ± 0.23	702.06 ± 0.63	758.00 ± 15.54	917.58 ± 67.10
	800	800.50 ± 0.25	801.86 ± 0.54	855.00 ± 14.48	1045.44 ± 82.43
	1000	900.32 ± 0.18	900.86 ± 0.35	933.00 ± 13.24	1054.48 ± 67.55
$M(\varepsilon) = N(\varepsilon)$ for FSP	50	1000.30 ± 0.22	1000.38 ± 0.23	1015.00 ± 8.55	1082.52 ± 55.61
	100	9.74 ± 1.29	29.13 ± 3.45	4.59 ± 0.34	2.27 ± 0.11
	200	6.63 ± 1.09	20.38 ± 2.82	3.83 ± 0.34	2.19 ± 0.10
	300	4.11 ± 0.78	8.45 ± 1.32	3.59 ± 0.36	2.04 ± 0.12
	400	3.22 ± 0.60	6.00 ± 1.02	2.78 ± 0.32	1.85 ± 0.13
	500	2.24 ± 0.31	4.10 ± 0.71	2.35 ± 0.27	1.73 ± 0.12
	600	1.72 ± 0.29	3.16 ± 0.54	1.94 ± 0.20	1.56 ± 0.11
	700	1.56 ± 0.21	2.38 ± 0.41	1.93 ± 0.24	1.42 ± 0.09
	800	1.29 ± 0.12	2.03 ± 0.32	1.58 ± 0.16	1.40 ± 0.10
	900	1.25 ± 0.13	1.93 ± 0.27	1.55 ± 0.14	1.44 ± 0.12
$M'(\varepsilon) = N'(\varepsilon)$ for SSP	1000	1.16 ± 0.09	1.43 ± 0.17	1.33 ± 0.13	1.21 ± 0.08
	50	1.15 ± 0.11	1.19 ± 0.12	1.15 ± 0.09	1.10 ± 0.06
	100	0.62 ± 0.09	2.45 ± 0.40	1.69 ± 0.27	12.00 ± 4.36
	200	0.73 ± 0.12	2.33 ± 0.37	1.45 ± 0.24	4.34 ± 1.22
	300	0.94 ± 0.15	1.74 ± 0.26	1.83 ± 0.26	3.20 ± 0.67
	400	1.15 ± 0.17	1.89 ± 0.29	1.75 ± 0.29	2.79 ± 0.54
	500	1.22 ± 0.13	1.85 ± 0.27	1.72 ± 0.24	2.45 ± 0.47
	600	1.25 ± 0.14	1.89 ± 0.25	1.68 ± 0.18	2.13 ± 0.34
	700	1.46 ± 0.16	1.84 ± 0.23	2.01 ± 0.26	2.15 ± 0.30
	800	1.43 ± 0.10	1.95 ± 0.22	1.83 ± 0.18	1.92 ± 0.21
Solution time (seconds)	900	1.65 ± 0.14	2.14 ± 0.22	2.07 ± 0.20	2.28 ± 0.27
	1000	1.70 ± 0.12	1.91 ± 0.16	1.96 ± 0.19	2.09 ± 0.24
	50	1.85 ± 0.15	1.92 ± 0.14	1.92 ± 0.15	1.96 ± 0.16

Table 3.2: Summary of results for APL1P.

To solve APL1P, we used the regularized decomposition code of Ruszczyński (1986)

Performance Measure	$n_0$	F:SRP(2)	F:A2RP(2)	F:A2RP(100)	S:A2RP
Sample sizes used:	50	150.50 ± 14.74	861.78 ± 57.90	2668.00 ± 149.02	6320.86 ± 512.06
	100	220.42 ± 20.02	837.42 ± 57.06	2707.00 ± 174.80	6277.70 ± 434.33
	200	317.16 ± 19.59	930.10 ± 55.09	2737.00 ± 177.92	6675.04 ± 423.62
	300	418.50 ± 19.97	916.32 ± 54.73	2788.00 ± 151.41	6705.56 ± 578.09
	400	495.60 ± 14.09	917.06 ± 42.79	2660.00 ± 159.81	6862.56 ± 478.47
	500	609.28 ± 15.45	1004.44 ± 53.35	2873.00 ± 165.94	6470.30 ± 485.73
	600	698.12 ± 16.09	1042.00 ± 46.25	2787.00 ± 153.60	6411.90 ± 456.25
	700	797.66 ± 15.24	1038.70 ± 36.82	2630.00 ± 150.88	6187.58 ± 404.08
	800	900.90 ± 13.62	1141.40 ± 35.11	2732.00 ± 144.54	6461.80 ± 418.29
	900	967.44 ± 12.14	1178.60 ± 32.25	2815.00 ± 157.86	6298.14 ± 403.58
1000	1076.60 ± 11.72	1236.08 ± 31.46	2741.00 ± 167.38	6297.38 ± 409.92	
Stopping iterations:	50	51.25 ± 7.37	406.89 ± 28.95	27.18 ± 1.49	3.77 ± 0.21
	100	61.21 ± 10.01	369.71 ± 28.53	27.07 ± 1.75	3.72 ± 0.19
	200	59.58 ± 9.79	366.05 ± 27.55	26.37 ± 1.78	3.99 ± 0.27
	300	60.25 ± 9.98	309.16 ± 27.37	25.88 ± 1.51	3.59 ± 0.23
	400	48.80 ± 7.05	259.53 ± 21.40	23.60 ± 1.60	3.63 ± 0.21
	500	55.64 ± 7.73	253.22 ± 26.67	24.73 ± 1.66	3.32 ± 0.21
	600	50.06 ± 8.04	222.00 ± 23.13	22.87 ± 1.54	3.36 ± 0.20
	700	49.83 ± 7.62	170.35 ± 18.41	20.30 ± 1.51	3.37 ± 0.24
	800	51.45 ± 6.81	171.70 ± 17.55	20.32 ± 1.45	3.43 ± 0.23
	900	34.72 ± 6.07	140.30 ± 16.13	20.15 ± 1.58	3.31 ± 0.23
1000	39.30 ± 5.86	119.04 ± 15.73	18.41 ± 1.67	3.08 ± 0.21	
Solution time (seconds)	50	0.95 ± 0.08	11.58 ± 1.62	4.69 ± 0.57	2.39 ± 0.25
	100	1.20 ± 0.15	10.34 ± 1.31	5.08 ± 0.73	2.27 ± 0.19
	200	1.41 ± 0.18	12.19 ± 1.49	5.21 ± 0.62	2.65 ± 0.27
	300	1.66 ± 0.24	11.54 ± 1.73	5.10 ± 0.56	2.52 ± 0.28
	400	1.57 ± 0.18	9.71 ± 1.11	4.74 ± 0.62	2.57 ± 0.23
	500	2.09 ± 0.25	12.27 ± 1.88	5.53 ± 0.67	2.41 ± 0.27
	600	2.33 ± 0.33	11.61 ± 1.82	5.07 ± 0.57	2.38 ± 0.23
	700	2.68 ± 0.39	9.21 ± 1.27	4.48 ± 0.58	2.39 ± 0.24
	800	3.12 ± 0.40	10.78 ± 1.43	4.73 ± 0.60	2.48 ± 0.25
	900	2.64 ± 0.44	9.91 ± 1.45	5.12 ± 0.56	2.36 ± 0.25
1000	3.59 ± 0.52	9.80 ± 1.48	4.91 ± 0.68	2.25 ± 0.20	

Table 3.3: Summary of results for 10D.

and Ruszczyński and Świetanowski (1997) and modified this code to allow for warm-starting when new samples are augmented. We note that warm-starting is more effective when sample size increases are small. This, coupled with the fact that F:SRP(2) stops in fewer number of iterations, results in fast solution times for F:SRP(2) for APL1P. F:A2RP(2) also utilizes warm-starting effectively, but for small values of  $n_0$  it tends to run for a long time. This is because A2RP estimators are more conservative (and hence more reliable for problems with undercoverage) than SRP estimators (Bayraksan and Morton, 2006, 2011). F:A2RP(100), on the other hand, runs for a fewer number of iterations, especially for small values of  $n_0$ . Even though larger sampling problems are solved, since a fewer number of problems are solved, solution times for F:A2RP(100) are smaller than those of F:A2RP(2). In SSP, the increases in sample sizes are much higher and the effect of warm-starting vanishes. Therefore, for APL1P, solution times of S:A2RP are generally

higher than those of F:A2RP(100). As the initial sample size increases, solution times of all procedures become similar.

Solution times for 10D, reported in Table 3.3, show a different pattern. We solved 10D using a dynamic programming approach with no warm-starting. Once again, the differences in solution times of F:SRP(2) and F:A2RP(2) are explained by differences in  $K(\varepsilon)$ . F:A2RP(100) improves on F:A2RP(2) by substantially decreasing  $K(\varepsilon)$ . Even though larger problems are solved per iteration, the decrease in the number of iterations reduces the overall solution time. This trend continues with S:A2RP, resulting in the lowest solution times for 10D despite the largest sample sizes.

We also note that the sample sizes used in the application of the sequential procedures for 10D are significantly smaller than theoretical estimates found in the literature. Let  $X^\varepsilon$  denote the set of  $\varepsilon$ -optimal solutions to (SP2) and let  $X_n^\varepsilon$  denote the set of  $\varepsilon$ -optimal solutions to (SP $_n$ ), where  $\varepsilon \geq 0$ . Note that, for  $\varepsilon = 0$ ,  $X^\varepsilon = X^*$  and  $X_n^\varepsilon = X_n^*$ . For a fixed significance level  $0 < \alpha < 1$  and  $\delta \in [0, \varepsilon)$ , an estimate of the sample size  $N$  required in order for the inequality  $\mathbb{P}(X_N^\delta \subset X^\varepsilon) \geq 1 - \alpha$  to hold is provided by Kleywegt et al. (2001) when  $|X|$  is finite:

$$N \geq \frac{3\sigma_{max}^2}{(\varepsilon - \delta)^2} \log \left( \frac{|X|}{\alpha} \right) = N_{LB}, \quad (3.1)$$

where

$$\sigma_{max}^2 = \max_{x \in X \setminus \{x^*\}} \text{var}[f(x^*, \tilde{\xi}) - f(x, \tilde{\xi})]. \quad (3.2)$$

The maximization problem in (3.2) can be very difficult to solve. Therefore, for 10D, we obtain a suitable lower bound on  $\sigma_{max}^2$ , which we denote by  $\hat{\sigma}_{max}^2$ , by taking advantage of the distributional properties (normal distribution) of the random weights and by eliminating the covariance term in the expansion of the variance through the use of the Cauchy-Schwartz inequality. Using  $\hat{\sigma}_{max}^2 \leq \sigma_{max}^2$ , a smaller lower bound on  $N$ ,  $\hat{N}_{LB} \leq N_{LB}$ , is obtained by

$$N \geq \frac{3\hat{\sigma}_{max}^2}{(\varepsilon - \delta)^2} \log \left( \frac{|X|}{\alpha} \right) = \hat{N}_{LB}. \quad (3.3)$$

Values of  $\hat{N}_{LB}$  for different values of  $\varepsilon$  and fixed values of  $\alpha$  and  $\delta$  are summarized in Table 3.4. To compute these results, we selected  $\delta = 0$  and  $\alpha = 0.10$  so that  $\sigma_{max}^2 \geq \hat{\sigma}_{max}^2 = 12664.1$ ; and  $\varepsilon$  is set so that  $X^\varepsilon = X^{dz^*}$ . As can be seen, the estimates of the

required sample size are prohibitively large for all practical purposes. For instance for 10D, a theoretical sample size  $N > 481,600$  is required to guarantee solutions  $x_N^*$  that are within 2% optimality with a 90% probability. This theoretical lower bound is nearly two orders of magnitude larger than the largest sample sizes reported in Table 3.3. It is well-known that such theoretical bounds are overly conservative; our results agree.

$d$	$\varepsilon$	$\hat{N}_{LB}$
0%	0.0400	2.1922E+008
1%	0.4267	1.9264E+006
2%	0.8534	4.8160E+005
3%	1.2801	2.1405E+005
4%	1.7068	1.2040E+005
5%	2.1335	7.7057E+004

Table 3.4: Lower bound on sample size  $N$  such that  $\mathbb{P}(X_N^\delta \subset X^\varepsilon) \geq 1 - \alpha$  for 10D.

### 3.4 Effect of $\varepsilon$

In this section, we examine the performance of the procedures as  $\varepsilon$  shrinks. To do this, we ran 100 replications of the procedures with varying values of  $\varepsilon = d \cdot z^*$  by decreasing  $d$  by an amount of 0.01 from 0.05 (5% of optimality) down to 0.01 (1% of optimality), and then further to  $d = 0.005$ ,  $d = 0.002$  and  $d = 0.001$  (0.1% of optimality). We set  $n_0 = 100$  and compared the performance of the procedures with respect to  $N(\varepsilon)$  and  $N'(\varepsilon)$ ,  $K(\varepsilon)$  and  $K'(\varepsilon)$ , and solution time. For brevity, we only report our results for APL1P as we obtained similar results for 10D. We also note that the values corresponding to  $d = 0.002$  are listed in Table 3.2.

Table 3.5 shows the results of our experiments as a percentage change in the stopping iteration, sample sizes used, and the solution time compared to the base case of  $d = 0.05$  for FSP and SSP with A2RP. There are substantial increases in the stopping iteration, sample sizes used, and solution time once we fall below  $d = 0.01$  (within 1% of optimality). The increases in  $N'(\varepsilon)$  and solution time are most substantial for SSP. As  $\varepsilon$  shrinks, F:A2RP(100) performs better relative to F:A2RP(2) in terms of solution time as fewer optimization problems are solved (see changes in  $K(\varepsilon)$ ). Propositions 4 and 6 indicate

asymptotic increases in  $N(\varepsilon)$  and  $N'(\varepsilon)$  proportional to  $1/\varepsilon^2$ , w.p.1. The smallest value of  $\varepsilon$  used in the experiments is  $\varepsilon = 24.6423$  when  $d = 0.001$ . This value of  $\varepsilon$  is still far from 0 for the asymptotic results to hold. Observe that the reason why we are able to obtain the results in Propositions 4 and 6 is that the optimality gap and variance estimate of the current candidate solution will be 0 for all sample sizes greater than a random  $N$  and  $M$ , based on the sample path, w.p.1 (Lemma 5). This will happen when all sampling problems return the same optimal solution. Figure 3.1 indicates that this probability is quite low even at higher sample sizes for APL1P. Therefore, a much smaller  $\varepsilon$  is needed to invoke these propositions for this problem.

We also examined how coverage probability changes as  $\varepsilon$  shrinks, depicted in Figure 3.4. F:A2RP(100) and S:A2RP have consistently high coverage. For FSP with SRP, coverage probability is high for large values of  $\varepsilon$  and as  $\varepsilon$  shrinks, it first decreases and then starts to increase again. We explain this as follows: When  $\varepsilon$  is large, as the results of §3.3 show, the probability of obtaining an  $\varepsilon$ -optimal solution is high. This results in good coverage with quick solution times. Then, as  $\varepsilon$  is decreased, this effect diminishes and the undercoverage problem of SRP shows. However, as  $\varepsilon$  is further decreased, the effect of Theorem 2 starts to be seen and the coverage improves. This trend can also be seen slightly in F:A2RP(2), although the coverage remains above 90%.

% Change in	Procedure	$d =$								
		0.001	0.002	0.005	0.01	0.02	0.03	0.04	0.05	
$N(\varepsilon), N'(\varepsilon)$	F:A2RP(2)	102.7	38.8	8.7	2.5	0.2	0.0	0.0	0.0	
	F:A2RP(100)	572.0	283.0	112.0	43.0	7.0	1.0	0.0	0.0	
	S:A2RP	9938.9	1804.8	277.7	54.4	3.1	0.2	0.0	0.0	
$K(\varepsilon), K'(\varepsilon)$	F:A2RP(2)	5135.0	1938.0	434.0	124.0	12.0	2.0	0.0	0.0	
	F:A2RP(100)	572.0	283.0	112.0	43.0	7.0	1.0	0.0	0.0	
	S:A2RP	132.0	119.0	98.0	49.0	7.0	1.0	0.0	0.0	
Sol. Time (s)	F:A2RP(2)	4557.2	1347.2	258.8	71.7	4.5	2.9	0.0	0.0	
	F:A2RP(100)	2390.9	826.2	223.4	67.1	12.4	0.7	0.2	0.0	
	S:A2RP	19069.8	2210.5	362.1	68.1	1.7	1.3	3.0	0.0	

Table 3.5: Change in sample size, stopping iteration, and solution time for APL1P

### 3.5 Selection of $\varepsilon$ and Results on 20TERM

In this section, we illustrate how to apply the procedures in practice on a large-scale test problem, 20TERM, whose solution is unknown. Recall that 20TERM has 40 independent stochastic parameters and  $1.1 \times 10^{12}$  scenarios. Based on our earlier findings, we set

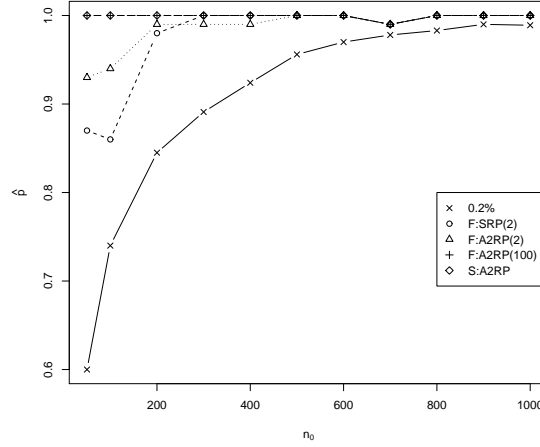


Figure 3.4: Coverage probability as  $\varepsilon$  shrinks.

$k_f^m = k_f^n = 3$  and again used  $\alpha = 0.10$  and  $h(n) = 1/\sqrt{n}$ . Next, we need to determine a value of  $\varepsilon$  to run the procedures. We proceed as follows.

Suppose the user is willing to use sample sizes of at most  $N_{\max}$  to solve the sampling approximation problems within the sequential procedures. We can use the stopping criterion (2.9) or (2.28) to estimate an appropriate value for  $\varepsilon$  that corresponds to  $N_{\max}$ . Rewriting the stopping rule yields the following lower bound on  $\varepsilon$ :

$$\varepsilon \geq G_{N(\varepsilon)} + \frac{t_{K(\varepsilon)} s_{N(\varepsilon)} + 1}{\sqrt{N(\varepsilon)}} = \varepsilon_{LB}. \quad (3.4)$$

An estimate  $\widehat{\varepsilon}_{LB}$  for  $\varepsilon_{LB}$  can be obtained by replacing  $N(\varepsilon)$  with  $N_{\max}$ , and estimating  $G_{N(\varepsilon)}$  and  $s_{N(\varepsilon)}^2$  through sampling as follows. For a moderate sample size  $N_{\text{trial}} < N_{\max}$ , we can run independent replications to collect values of  $G_{N_{\text{trial}}}$  and  $s_{N_{\text{trial}}}^2$ . We can then use the averages of these values collected, denoted  $\bar{G}_{N_{\text{trial}}}$  and  $\bar{s}_{N_{\text{trial}}}^2$ , as estimates for  $G_{N(\varepsilon)}$  and  $s_{N(\varepsilon)}^2$ , respectively in (3.4). We select  $\varepsilon = \widehat{\varepsilon}_{LB}$ , where

$$\widehat{\varepsilon}_{LB} = \bar{G}_{N_{\text{trial}}} + \frac{z_{\alpha} \bar{s}_{N_{\text{trial}}} + 1}{\sqrt{N_{\max}}}$$

For 20TERM, we used  $N_{\max} = 750$  and calculated  $\bar{G}_{N_{\text{trial}}}$  and  $\bar{s}_{N_{\text{trial}}}$  by taking the averages of 25 replications with  $n_0 = N_{\text{trial}} = 500$ . Under these experimental settings, we obtained  $\varepsilon = \widehat{\varepsilon}_{LB} = 50.61$ . Using this value for  $\varepsilon$ , we ran the procedures on 20TERM and report the results in Table 3.6.

Performance Measure	$n_0$	F:A2RP(2)	F:A2RP(100)	S:A2RP
Sample sizes used:	500	502.48 $\pm$ 0.64	593.00 $\pm$ 21.21	683.90 $\pm$ 46.76
Stopping iterations:	500	2.24 $\pm$ 0.32	1.93 $\pm$ 0.21	1.54 $\pm$ 0.11
Coverage probability estimate:	500	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
Solution time (seconds):	500	1906.83 $\pm$ 208.30	2104.78 $\pm$ 312.32	1794.30 $\pm$ 214.36

Table 3.6: Summary of results for 20TERM

In Table 3.6, we report 90% confidence intervals on the sample sizes used, the stopping iterations, and the solution time computed out of 100 independent runs. The procedures stop on average in a small number of iterations using moderate sample sizes, resulting in relatively tight confidence intervals. In particular, SSP stops in slightly fewer number of iterations and faster overall solution time. The average solution times illustrate the increased difficulty in solving sampling instances of 20TERM compared to the two smaller test problems APL1P and 10D considered earlier.

To test the quality of the solutions obtained, we conducted a separate study to estimate their optimality gaps. We used the *multiple replications procedure* (MRP) of Mak et al. (1999), which has been shown to be more conservative compared to A2RP in empirical studies (Bayraksan and Morton, 2006). That is, MRP typically results in larger CI widths than A2RP with more than the desired coverage probability. We generated 30 gap estimators, each using a sample size of 500, calculated their variance and formed a 90% confidence interval on the optimality gap of the solutions obtained. Note that these 30 samples of size 500 are independent of the ones used for estimating  $\varepsilon$  above. We checked the frequency of the MRP interval widths that were below  $\varepsilon$ . The row ‘‘Coverage probability estimate’’ in Table 3.6 shows the results. Even though MRP CIs can be more conservative, we found that all MRP CI widths were within  $\varepsilon$ . This suggests that we are 90% confident that our solutions are indeed  $\varepsilon$ -optimal solutions. To further see how these solutions compare to an actual optimum solution, we calculated a lower bound on  $z^*$  by solving a single scenario problem where all the stochastic parameters are set to their expected values, called the EV-problem. Since the second-stage decisions are continuous, the optimum value of the EV-problem results in a valid lower bound on  $z^*$ . We found  $z_{EV}^* = 239,272.8 \leq z^*$ . Comparing  $z_{EV}^*$  to the  $\varepsilon$  used, we see that the  $\varepsilon$ -optimal solutions are indeed within at most 0.0211% of optimality. The results indicate that we have obtained high-quality  $\varepsilon$ -optimal solutions with a high probability (approximately 90%).

To further verify this method of selecting  $\varepsilon$ , we also tested it on APL1P and obtained favorable results. For instance, with  $n_0 = N_{\text{trial}} = 100$  and  $N_{\text{max}} = 1000$ , we obtain an  $\varepsilon$  that is 0.52% of  $z^*$ . If the user wishes to solve smaller problems, e.g.,  $N_{\text{max}} = 500$ ,  $\varepsilon$  is slightly increased to 0.59%. For brevity, we do not report on the details of these experiments. However, Tables 3.2 and 3.5 can be used to see that the procedures performed well with this  $\varepsilon$  selection for APL1P.

Finally, we tested the effect of the choice of the inflation function  $h(n)$ . The results (sample sizes used, stopping iteration, coverage probability) remain virtually the same for the following choices of  $h(n)$ :  $1/n$ ,  $1/\sqrt{n}$ ,  $1/\log n$ , and  $1/\log \log n$ . Notice that for these test problems,  $\varepsilon$  typically takes on values that are much greater than the largest value of  $h(n)$ , and the procedures stop when both gap and variance estimators are relatively quite small; therefore, the choice of  $h(n)$  does not have a significant effect.

### 3.6 Summary and Concluding Remarks

In this section, we summarize our observations and provide preliminary guidelines for implementation based on our computational experiments.

- While increasing the sample sizes, we can choose to augment previously generated observations or generate an entirely new set of independent observations (resampling). Generating completely new independent observations results in quicker solution times but can lower coverage probability, as has been observed earlier for a different stopping rule (Bayraksan and Morton, 2011).
- The computational experiments indicate that when only the samples used to evaluate the candidate solutions are resampled (and the samples for generating candidate solutions are always augmented), the process somewhat adapts itself to the stream of candidate solutions generated. That is, if a bad solution stream is obtained, then there is a high likelihood that the procedure stops at a bad solution, the same is true for a good solution stream. This can result in a less than desired coverage probability at small initial sample sizes. On the other hand, if both sample streams—the one

to generate candidate solutions and the one to evaluate them—are resampled, this effect diminishes. In this case, the procedures stop quicker and interestingly, the coverage probability remains high. Therefore, we recommend to resample an entirely new set of observations for both sample streams. We resample at a frequency of every 3 iterations, as this resulted in the quickest solution times with desired coverage levels.

- SRP stops faster than A2RP but can have low coverage for small initial sample sizes. This observation matches with our earlier observations in the nonsequential and sequential settings (Bayraksan and Morton, 2006, 2011). It is well known that a valid interval estimator in the nonsequential setting can result in a lower coverage probability when used in a sequential setting (Glynn and Whitt, 1992). Therefore, we recommend the use of more conservative but more reliable A2RP.
- Computationally, larger increases in sample sizes (e.g., 100 by 100 as opposed to 2 by 2) can be preferable. Larger increases in sample sizes result in fewer iterations; fewer optimization problems are solved in this case. Even though larger problems are solved, when considerably fewer of them are solved, overall solution times decrease.
- When the initial sample size is small to have reliable estimators and/or when there is a large bias or variance, SSP can yield large sample size estimates. This can be computationally taxing. Therefore, when the initial sample size is small, we recommend the use of FSP with larger increases in sample sizes. When the initial sample size is large enough (typically  $\geq 500$ ), SSP can result in moderate jumps with faster overall solution time.
- Theorems 2 and 3 suggest selecting  $\varepsilon$  as small as possible. However, as the experiments in §3.4 as well as the results of Propositions 4 and 6 show, the sample sizes when the procedures stop with a small  $\varepsilon$  can be quite large; hence, this can be computationally demanding. Our experiments indicate that with the use of *reliable* confidence interval estimators, high-quality solutions can be obtained with a high

probability when moderate values of  $\varepsilon$  are used. In §3.5, we provide an approach to estimate  $\varepsilon$  by inverting the stopping rule and calculating an estimate for  $\varepsilon$  as a function of the targeted sample size. Empirical results indicate that this approach performs well in practice.

## CHAPTER 4

## SAMPLING-BASED SEQUENTIAL APPROXIMATION METHOD

In this chapter, we develop an algorithm for two-stage stochastic programming of the form of (SP2) with convex second stage and with uncertainty in the right-hand side. The algorithm draws on techniques from bounding and approximation methods as well as sampling-based approaches. In particular, we sequentially refine a partition of the support of the random vector and, through Jensen's inequality, generate deterministically-valid lower bounds on the optimal objective function value. An upper bound estimator is formed through a stratified Monte Carlo sampling procedure that includes the use of a control variates variance reduction scheme. We present stopping rules that ensure an asymptotically valid confidence interval for the quality of the proposed solution. There are three key differences between this algorithm and the fixed-width sequential sampling algorithms developed in Chapter 2, as outlined below:

- (i) *Candidate Solution Generation.* While the candidate solutions for the algorithms in Chapter 2 were generated by solving a sequence of sampling approximate problems ( $SP_n$ ) with increasing sample sizes, for the algorithm developed in this chapter, candidate solutions are generated by solving a lower bounding approximate problem derived from Jensen's inequality as the support of the random vector is divided into further refined partitions.
- (ii) *Lower Bound.* Both the lower bounds and the upper bounds in gap estimation are obtained through sampling. For the algorithms in Chapter 2 only the upper bounds are sampling-based, whereas in this chapter, the lower bounds are deterministic and are obtained as the optimal value of the Jensen's-based lower bounding approximate problem used in (i) to generate the candidate solutions.
- (iii) *Stopping Rule.* The procedures developed in Chapter 2 stop when the width of the confidence interval on the optimality gap plus an inflation factor falls below a

*fixed* pre-specified value  $\varepsilon$  and the confidence interval output on the optimality gap when the procedure stops is fixed,  $[0, \varepsilon]$ . Thus, they are *fixed-width*. The algorithm developed in this chapter, however, has a *relative-width* stopping rule. The sequential procedure stops when the point estimate on the optimality gap falls below a pre-specified proportion relative to its sampling standard deviation and the confidence interval output on the optimality gap of the candidate solution when the procedure stops is of the form  $[0, h \cdot s_n]$  where  $h > 0$  is a constant and  $s_n$  is the sample standard deviation of the optimality gap estimate. Thus, the larger the variance, the larger the quality statement on the solutions obtained.

In light of these key differences, both the computational considerations and theoretical analysis of the sampling-based sequential approximation method developed in this chapter require fundamentally different treatments than the algorithms in Chapter 2.

This chapter is organized as follows. In the next section, we review relevant literature on sequential approximation and related methods. In §4.2, we discuss the class of problems to which the procedure applies. In §4.3, we review the classical sequential approximation method (SAM) for two-stage stochastic programs, from which the sampling-based sequential approximation method (SSAM) is inspired. In §4.4, we present the sampling-based sequential approximation method. We establish the asymptotic validity and finite stopping of SSAM in §4.5. We conclude this chapter with a summary in §4.6. We present computational experiments in Chapter 5.

#### 4.1 Literature Review

We develop an algorithm for approximately solving a class of two-stage stochastic programs by combining Jensen's inequality (Jensen, 1906) with a Monte Carlo estimator. Our approach is related to a classic approximation scheme in stochastic programming called the *sequential approximation method* (SAM) (Edirisinghe and Ziemba, 1992; Frauendorfer, 1992; Huang et al., 1977; Kall et al., 1988). A SAM uses upper and lower bounds on the expected value of the future-cost function, i.e.,  $\mathbb{E}[Q(x, \xi)]$  in (SP2). These bounds exploit convexity and, in this classic approach, are iteratively refined. Our ap-

proach is similar except that we replace the more computationally expensive of these two bounds with a Monte Carlo estimator.

Despite wide applicability, two-stage stochastic programs suffer from a curse in the dimension of the random parameters. When the dimension of the random vector is large, the expected value of the future-cost function is a high-dimensional integral, which is usually computationally intractable, even when the first-stage decision vector is fixed. As a result, various approximation schemes have been developed.

SAM, a classic approximation scheme, calculates deterministically-valid upper and lower bounds on the objective function. For a two-stage stochastic program, under certain convexity properties, lower bounds are typically based on Jensen's inequality. While there are a number of exceptions (see, e.g., (Birge and Dulá, 1991; Birge and Teboulle, 1989; Birge and Wets, 1989; Kall, 1991; Morton and Wood, 1999)), many of the upper bounding schemes are rooted in the Edmundson-Madansky inequality (Madansky, 1959, 1960). In contrast to Jensen's lower bound, which requires only one function evaluation for each element of the partition, the number of function evaluations required to apply the Edmundson-Madansky inequality to a random vector with independent components can grow exponentially with the dimension of the random vector. We can avoid this exponential growth by using simplicial or polyhedral bounding domains in place of hyper-rectangles, albeit with weakened bounds; see, e.g., Dupačová (1976); Edirisinghe (1996); Frauendorfer (1992); Gassmann and Ziemba (1986). Still, generating sufficiently tight upper bounds is usually the computational bottleneck in a SAM.

There is a large literature on forming upper and lower bounds on the objective function of such stochastic programs (Birge and Wets, 1987; Dokov and Morton, 2005; Dupačová, 1987, 1966; Kall, 1991; Morton and Wood, 1999; Topaloglu, 2009), and also on using these bounds in an algorithmic way to find near-optimal solutions (Birge and Wallace, 1986; Birge and Wets, 1986; Edirisinghe and Ziemba, 1992; Frauendorfer, 1992; Frauendorfer and Kall, 1988; Huang et al., 1977). These SAMs progressively produce tighter bounds by refining a partition of the random vector's support and employing these bounds conditionally on each cell of the partition.

Monte Carlo sampling provides another popular means to approximately solve a

stochastic program. In simplest form, a sampling-based approximation replaces probabilistic statements that appear in a stochastic program with sampling-based counterparts. For example, we can replace an expectation by the corresponding sample mean. When using Monte Carlo sampling to find an approximate solution, the approximating problem can be solved with any solution procedure appropriate for the underlying problem. As such, this approximation method is attractive when the dimension of the random vector is large and the distribution has many realizations, possibly infinite. Asymptotic properties of sampling-based approximations are studied extensively in the literature; see, e.g., Attouch and Wets (1981); Dupačová and Wets (1988); Homem-de-Mello (2008); King and Rockafellar (1993); Shapiro (1991) and the survey by Shapiro (2003). There is also a rich literature on sampling-based algorithms that modify deterministic algorithms, replacing the function values and/or (sub)gradients that are required throughout the algorithm with Monte Carlo estimators. One such class of algorithms involves stochastic versions of the steepest-descent algorithm in which the subgradients used within the algorithm are estimated via Monte Carlo sampling; see, e.g., Ermoliev (1988); Gaivoronski (1988); Kushner and Yin (1997); Nemirovski et al. (2009); Robbins and Monro (1951). Another important class of such algorithms is rooted in the L-shaped method to solve stochastic programs with recourse, where the cutting planes are estimated via sampling (Dantzig and Glynn, 1990; Higle and Sen, 1991b, 1994, 1996b; Infanger, 1992).

In this chapter, we develop a method that combines deterministically valid lower bounds on the objective function, those used by a SAM through Jensen's inequality, with Monte Carlo sampling-based upper bound estimates. We use this approach within a sequential framework to devise an algorithm that generates high-quality solutions with high probability. The result is a new algorithm that we call the *sampling-based sequential approximation method* (SSAM). Valid termination of an optimization algorithm with noisy bounds on the optimality gap demands attention to accompanying sequential issues. We describe how to deal with those issues, and we further propose a variance-reducing estimator. This estimator exploits special structures associated with the iterative refinement scheme and as a result, the magnitude of variance reduction tends to grow as the algorithm converges. To devise rules to increase the sample sizes and to stop SSAM, we adapt the

sequential sampling framework of Bayraksan and Morton (2011). This framework takes as input a sequence of solutions with optimal limit points and evaluates them by forming statistical estimators of their optimality gaps. The procedure sequentially increases the sample size and stops when the optimality gap estimate falls below a fraction of the sample standard deviation. The sequential sampling procedure of Bayraksan and Morton (2011) applies to a broad class of algorithms like SAM. However, instead of using SAM simply to generate the sequence of candidate solutions that are used as input to the sequential framework, we integrate the two algorithms, exploiting the advantages of each.

SSAM integrates the two algorithmic frameworks as follows. First, it uses SAM to generate the sequence of candidate solutions. That is, it sequentially refines a partition of the random vector's support, and employs Jensen's inequality conditionally on each cell of the partition, to generate a candidate solution. Next, SSAM uses Monte Carlo sampling-based upper bounds, alleviating the computational bottleneck caused by calculating Edmundson-Madansky upper bounds in SAM. Calculating the upper bound via Monte Carlo sampling corresponds to estimating the objective function value for the current candidate solution, which is the usual sample mean estimator. We reduce the variance of this estimator within SSAM using stratified sampling and control variates. Finally, the procedure iteratively increases the sample size and terminates according to the sequential sampling framework of Bayraksan and Morton (2011). SSAM requires unique changes to the theory presented by Bayraksan and Morton (2011) and we present these in §4.5.

SSAM uses Jensen's lower bound applied to each cell of the partition and then weighted by the associated probability mass and summed over all cells. We refer to this bound simply as Jensen's lower bound. An alternative way is to calculate lower bounds via Monte Carlo sampling. This requires solution of an independent sampling problem, which is the computational bottleneck when estimating optimality gaps via sampling (Bayraksan and Morton, 2006; Mak et al., 1999; Norkin et al., 1998). While Jensen's lower bound requires modest effort to compute, this approach to lower bounding leads to three potential drawbacks. First, the convexity assumption required to use Jensen's inequality restricts the class of problems to which SSAM applies (see next section). Second, for some problems, Jensen's lower bound can be loose. Third, because

SSAM forms a partition of the random vector's support, just as in SAM, it can be less effective as the dimension of the random vector grows. The use of tighter lower bounds (Dokov and Morton, 2005; Topaloglu, 2009), through means other than refining the partition, may improve SSAM's performance in these cases, although we do not pursue such alternatives here. In our computational results, we describe a problem with moderate dimension for which the Jensen lower bound is reasonably tight with a manageable number of cells in the partition. We test this numerically in Chapter 5 and find that SSAM performs reasonably well.

## 4.2 Problem Class

We consider two-stage stochastic programs of the form:

$$z^* = \min_{x \in X} \{f(x) \equiv c(x) + \mathbb{E}[Q(x, \xi)]\}, \quad (\text{SP2})$$

where

$$\begin{aligned} Q(x, \xi) = \min_{y \geq 0} \quad & q(y) \\ \text{s.t.} \quad & g(y) \leq h(\xi) - T(x, \xi). \end{aligned} \quad (4.1)$$

Recall that in Chapter 1 we made the following assumptions. First, we assume the random vector  $\xi$  has finite dimension which we denote by  $d_\xi$ , and known distribution, which we assume independent of decision  $x$ . As a reminder, we denote the set of feasible first-stage decisions by  $X \subseteq \mathbb{R}^{d_x}$ . Here,  $c : X \rightarrow \mathbb{R}$ ,  $Q : X \times \text{co}(\Xi) \rightarrow \mathbb{R}$ ,  $q : \mathbb{R}^{d_y} \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^{d_y} \rightarrow \mathbb{R}^{m_y}$ ,  $h : \text{co}(\Xi) \rightarrow \mathbb{R}^{m_y}$ , and  $T : X \times \text{co}(\Xi) \rightarrow \mathbb{R}^{m_y}$ , where  $\Xi$  is  $\xi$ 's support and  $\text{co}(\cdot)$  denotes the convex hull operator. We also assume (from Chapter 1) that the expectation in (SP2) is finite for all  $x \in X$ ; model (4.1) has a finite optimal solution achieved by a feasible  $y$  for every element of  $X \times \text{co}(\Xi)$  (relatively complete recourse); and, model (SP2) has a finite optimal solution achieved on  $X$ . Recall that these are assumptions (A1)-(A3) from Chapter 1. In the sampling-based sequential approximation method, we make the following additional assumptions with respect to (SP2):

(A8)  $Q(x, \cdot)$  is convex on  $\text{co}(\Xi)$  for all  $x \in X$ ; and,

(A9)  $\xi$  has independent components, and  $h(\cdot)$  and  $T(x, \cdot)$  are affine on  $\mathbb{R}^{d_\xi}$  for all  $x \in X$ .

Assumption (A8) allows us to use Jensen's inequality to form lower bounds on the objective function. Under assumption (A9), a sufficient condition to ensure that assumption (A8) holds is that  $q(\cdot)$  and  $g(\cdot)$  are convex functions; see, e.g., Fiacco and Kyparisis (1986). Given these conditions, convexity of  $Q(\cdot, \xi)$  on  $\text{co}(X)$  holds, provided  $T$  is convex on  $\text{co}(X)$  for each fixed element of  $\text{co}(\Xi)$ . From the perspective of the underlying probability model, assumption (A9), with the affine dependence of  $h(\cdot)$  and  $T(x, \cdot)$  on  $\xi$ , permits capturing first order notions of dependency of the random parameters in (SP2), such as those that arise in commonly-used linear factor models.

With these assumptions in place, model (SP2) can take a variety of forms. For instance, it can be a stochastic linear program or a stochastic convex program or  $X$  can have integrality restrictions, leading to a stochastic integer program. This changes the requisite tools for solving the lower bounding problems (see §§4.3-4.4 for details) but the algorithmic framework of SSAM that we put forward remains the same.

Two-stage stochastic linear programs (SLP-2s) are widely studied in the literature. These are a special case of the problem class we consider, at least when the stochastic parameters appear only on the right-hand side of the second stage linear program. Our computational experiments include such SLP-2s. In an SLP-2,  $c(x) = \bar{c}x$  for an  $d_x$ -dimensional row vector  $\bar{c}$ ,  $q(y) = \bar{q}y$  for an  $d_y$ -dimensional row vector  $\bar{q}$ ,  $g(y) = Wy$  for an  $m_y \times d_y$  matrix  $W$ ,  $T(x, \xi) = \bar{T}(\xi)x$ , where the  $m_y \times d_x$  matrix  $\bar{T}(\xi)$  is affine in  $\xi$ , and  $X$  is polyhedral. So, for an SLP-2,  $f(x) = \bar{c}x + \mathbb{E}[Q(x, \xi)]$ , where

$$Q(x, \xi) = \min_{y \geq 0} \{ \bar{q}y : Wy = h(\xi) - \bar{T}(\xi)x \}. \quad (4.2)$$

When  $\Pi = \{ \pi : \pi W \leq \bar{q} \} \neq \emptyset$  and is bounded,  $Q$  is finite on  $X \times \text{co}(\Xi)$ . This coupled with a nonempty and compact  $X$  and the existence of relevant expectations ensures that SLP-2 has a finite optimal solution. As indicated above, with stochasticity only on the right-hand side of the constraints in (4.2), under assumption (A9) we have that  $Q(x, \cdot)$  is convex on  $\text{co}(\Xi)$ . We discuss in the next section how this enables use of the inequalities of Jensen and Edmundson-Madansky.

In addition to (A8) and (A9), we make the following assumption:

$$(A10) \sup_{x \in X} \text{var} [Q(x, \xi)] < \infty.$$

We will be using sampling and to be able to invoke an appropriate form of the central limit theorem, we need to have, at a minimum, finite variances. Assumption (A10) guarantees this minimal requirement.

### 4.3 Review of Sequential Approximation Method

Significant parts of the algorithmic machinery from the literature on sequential approximation methods are inherited by our SSAM, and so we review the key ideas behind a SAM here. For more details on SAMs, see, e.g., Kall et al. (1988). Let  $a_i < b_i, i = 1, 2, \dots, d_\xi$ , be such that  $\Xi \subseteq \prod_{i=1}^{d_\xi} [a_i, b_i]$ , where we allow for the possibility that  $a_i = -\infty$  and/or  $b_i = \infty$ .

#### 4.3.1 Bounding the Objective Function

We now review the lower bounding function on  $f(x)$  that arises from Jensen's inequality applied to a partition of  $\Xi$ . Let  $\mathcal{S} = \{\Xi_\ell : \ell = 1, 2, \dots, \nu\}$  denote a partition of  $\Xi$ ; that is,  $\Xi_i \cap \Xi_j = \emptyset, i \neq j$  and  $\bigcup_{\ell=1}^{\nu} \Xi_\ell = \Xi$ . Let  $p^\ell = \mathbb{P}(\xi \in \Xi_\ell)$  denote the probability mass of cell  $\ell$  and let  $\bar{\xi}^\ell = \mathbb{E}[\xi | \xi \in \Xi_\ell]$  denote  $\xi$ 's mean, conditional on being in cell  $\ell$ . Then,  $\mathbb{E}[Q(x, \xi)] = \sum_{\ell=1}^{\nu} p^\ell \mathbb{E}[Q(x, \xi) | \xi \in \Xi_\ell]$ . Since  $Q(x, \cdot)$  is convex on  $\text{co}(\Xi_\ell)$ , application of Jensen's inequality on each cell results in a lower bounding function induced by the partition  $\mathcal{S}$ ,

$$L_{\mathcal{S}}(x) \equiv c(x) + \sum_{\ell=1}^{\nu} p^\ell Q(x, \bar{\xi}^\ell) \leq f(x). \quad (4.3)$$

In what follows, we assume that each cell of the partition has form  $\Xi_\ell = \prod_{i=1}^{d_\xi} [a_i^\ell, b_i^\ell]$ , where  $a_i \leq a_i^\ell < b_i^\ell \leq b_i, \ell = 1, 2, \dots, \nu$ . Under assumption (A8), this means that forming  $p^\ell$  and  $\bar{\xi}^\ell$  only requires computing univariate expectations.

Let  $U_{\mathcal{S}}(x)$  denote a deterministically-valid upper bound that, like the Jensen bound, is adapted to the partition  $\mathcal{S}$ , and has the following form:

$$U_{\mathcal{S}}(x) \equiv c(x) + \sum_{\ell=1}^{\nu} p^\ell \mathbb{E}[Q(x, \xi_{U_\ell})] \geq f(x). \quad (4.4)$$

In the Edmundson-Madansky bound,  $\xi_{U_\ell}$  is a random variable whose support is the set of extreme points of  $\Xi_\ell$ , and so when  $\Xi_\ell = \prod_{i=1}^{d_\xi} [a_i^\ell, b_i^\ell]$  this means evaluating  $\mathbb{E}[Q(x, \xi_{U_\ell})]$  requires  $2^{d_\xi}$  function evaluations. In our description of a SAM, we skimp on the details of Edmundson-Madansky upper bounds because SSAM does not use such bounds. See, for example, Frauendorfer (1992); Kall et al. (1988) for further details on this conditional version of the Edmundson-Madansky inequality. We do note that the simplest Edmundson-Madansky upper bounds require bounded support, i.e.,  $a_i < \infty$  and  $b_i < \infty$  for all  $i = 1, 2, \dots, d_\xi$ , although related upper bounds of Birge and Wets (1987) and Edirisinghe and Ziemba (1994) allow for unbounded support, provided a certain sublinearity property of  $Q(x, \cdot)$  holds as its second argument grows large. Our sampling-based upper bounds also allow for unbounded support and do not require sublinearity.

#### 4.3.2 Generating Solutions

Given a partition  $\mathcal{S}$ , we generate a candidate solution,  $\hat{x}$ , in SAM by optimizing  $L_{\mathcal{S}}(x)$  over  $X$ :

$$\hat{x} \in \underset{x \in X}{\operatorname{argmin}} L_{\mathcal{S}}(x). \quad (4.5)$$

For a given  $x \in X$ , recall that we let  $\mu_x$  denote its optimality gap,  $\mu_x = f(x) - z^*$ . For a candidate solution  $\hat{x}$  found by (4.5), we can bound  $\mu_{\hat{x}}$  via  $\mu_{\hat{x}} \leq U_{\mathcal{S}}(\hat{x}) - L_{\mathcal{S}}(\hat{x})$ . This follows immediately from (4.3) and (4.4) and the fact that  $\hat{x}$  minimizes the lower bound. SAM sequentially refines the partition  $\mathcal{S}$  to improve the candidate solution as well as the bound on the optimality gap. The procedure stops when the upper bound on  $\mu_{\hat{x}}$  is sufficiently small (see §4.3.4 for details).

#### 4.3.3 Refining Partitions

Three issues arise when refining a partition  $\mathcal{S} = \{\Xi_\ell : \ell = 1, 2, \dots, \nu\}$ . First, which cell(s) of the partition should we refine? Second, given a cell, perpendicular to which axis should we construct a hyperplane to form two new cells? Third, at which point on the designated axis should the splitting hyperplane be positioned? In SAM, and in SSAM of the next section, we answer these questions for a fixed  $\hat{x}$  found via (4.5). We sketch basic

ideas here and leave it to §4.4.4 to detail specifics for SSAM.

To answer the first question, note that the error bound  $U_{\mathcal{F}}(\hat{x}) - L_{\mathcal{F}}(\hat{x})$  on  $\mu_{\hat{x}}$  may be expressed as a sum of contributions from each cell. The contribution of cell  $\ell$  is given by:

$$p^\ell \left[ \mathbb{E}[Q(\hat{x}, \xi_{U_\ell})] - Q(\hat{x}, \bar{\xi}^\ell) \right]. \quad (4.6)$$

We refine those cells with the largest errors, as defined by (4.6).

If  $Q(\hat{x}, \cdot)$  is linear on  $\Xi_\ell$  then the inequality,  $Q(\hat{x}, \bar{\xi}^\ell) \leq \mathbb{E}[Q(\hat{x}, \xi)]$ , holds with equality. The same statement holds for the inequality  $\mathbb{E}[Q(\hat{x}, \xi)] \leq \mathbb{E}[Q(\hat{x}, \xi_{U_\ell})]$  under the Edmundson-Madansky bound based on convexity, and arguably should also hold for any reasonable bound. Unless  $Q(\hat{x}, \cdot)$  is a polyhedral function, we cannot expect to achieve linearity, even with very fine partitions. Still, this observation guides answers to the second question in that the greater the curvature of the function  $Q(\hat{x}, \cdot)$ , the larger the error associated with a cell.

Assume that cell  $\ell$  is to be split. Jensen's inequality arises from the best linear approximation of  $Q(\hat{x}, \cdot)$  across a cell (see, e.g., §3.4.1 of (Kall and Wallace, 1994)), which is the first-order Taylor approximation formed at the conditional mean of the cell,  $\bar{\xi}^\ell$ . Assume for the moment that  $Q(\hat{x}, \cdot)$  is twice differentiable (as is our computational example—the asset allocation problem—in Chapter 5). Forming a second-order Taylor approximation of  $Q(\hat{x}, \cdot)$  at  $\bar{\xi}^\ell$  along the  $i$ th coordinate axis, computing the expected value of the difference with the first-order approximation, and selecting the axis with the largest such expected error means that we choose axis:

$$i^* \in \operatorname{argmax}_{i=1, \dots, d_\xi} \left[ \frac{\partial^2 Q(\hat{x}, u)}{\partial u_i^2} \Big|_{u=\bar{\xi}^\ell} \operatorname{var}(\xi_i | \xi \in \Xi_\ell) \right]. \quad (4.7)$$

In the case of SLP-2, for example,  $Q(\hat{x}, \cdot)$  is not even once differentiable and other approaches are required. Birge and Wets (1986) suggest using differences of subgradients at the endpoints of each edge of a bounded cell to make this determination. Frauendorfer and Kall (1988) found that examining differences between first-order Taylor approximations and actual recourse function values along each edge often further improves the refinement procedure. For the  $\ell^{\text{th}}$  cell, this latter approach means we can select

the axis to split via

$$i^* \in \operatorname{argmax}_{i=1, \dots, d_\xi} \min \left\{ Q(\hat{x}, a^\ell) - [Q(\hat{x}, v_{i+}^\ell) + \pi(\hat{x}, v_{i+}^\ell)(a^\ell - v_{i+}^\ell)], \right. \\ \left. Q(\hat{x}, v_{i+}^\ell) - [Q(\hat{x}, a^\ell) + \pi(\hat{x}, a^\ell)(v_{i+}^\ell - a^\ell)] \right\}, \quad (4.8)$$

where  $a^\ell = (a_1^\ell, \dots, a_{d_\xi}^\ell)$ ,  $v_{i+}^\ell = a^\ell + (b_i^\ell - a_i^\ell)e_i$ ,  $e_i$  is the  $i^{\text{th}}$  unit vector, and  $\pi(\hat{x}, \cdot)$  is a subgradient of  $Q(\hat{x}, \cdot)$ . The axis selection criterion (4.8) requires solving a second-stage problem at  $d_\xi + 1$  of the corner points of a bounded cell.

Finally, we turn to the third question. Once the axis to split,  $i^*$ , has been determined, the corresponding component of the conditional mean may be used as the point to position the splitting hyperplane (Birge and Wets, 1986; Kall et al., 1988).

#### 4.3.4 SAM and Convergence Properties

Algorithm 1 summarizes the discussion in §§4.3.1-4.3.3, providing a brief algorithmic statement of SAM. Note that this algorithm follows the same framework as outlined in Chapter 1, and the fixed-width sequential algorithms of Chapter 2. Going forward, we use index  $k$  to denote an iteration of an algorithm. So,  $\mathcal{S}_k$  denotes the partition at iteration  $k$  of SAM, which contains  $v_k$  cells, and we denote the corresponding candidate solution, obtained in (4.5) with  $\mathcal{S} = \mathcal{S}_k$ , by  $\hat{x}_k$ .

We now turn to the limiting behavior of the SAM as  $k \rightarrow \infty$ . Let

$$f_k(x, \xi) = c(x) + \sum_{\ell=1}^{v_k} \mathbb{I}(\xi \in \Xi_\ell) \left[ Q(x, \bar{\xi}^\ell) + \nabla_\xi Q(x, \bar{\xi}^\ell)(\xi - \bar{\xi}^\ell) \right],$$

where we use  $\nabla_\xi Q(x, \bar{\xi}^\ell)$  to denote a (sub)gradient of  $Q(x, \cdot)$  at  $\bar{\xi}^\ell$  and  $\mathbb{I}(\cdot)$  denotes the indicator function. Note that  $L_{\mathcal{S}_k}(x) = \mathbb{E}[f_k(x, \xi)]$ .

In this way,  $f_1(x, \xi)$  is integrable with  $\mathbb{E}f_1(x, \xi) = c(x) + Q(x, \mathbb{E}\xi)$ . And, because the partition for  $k + 1$  is a refinement of that at  $k$ , we have  $f_k(x, \xi) \leq f_{k+1}(x, \xi) \leq c(x) + Q(x, \xi)$ . Thus, if we refine the partitions in such a way to ensure pointwise convergence

$$\lim_{k \rightarrow \infty} f_k(x, \xi) = c(x) + Q(x, \xi), \quad (4.9)$$

almost surely (a.s.), then we can employ the monotone convergence theorem to infer

$$\lim_{k \rightarrow \infty} \mathbb{E}[f_k(x, \xi)] = c(x) + \mathbb{E}[Q(x, \xi)]. \quad (4.10)$$

---

**Algorithm 1: Sequential Approximation Method (SAM)**


---

- step 0 (Initialization)**  
 | select  $\varepsilon > 0$  and let  $k = 1$ ,  $\mathcal{S}_k = \{\Xi\}$  and  $v_k = |\mathcal{S}_k|$ ;
- step 1 (Generate candidate solution)**  
 | let  $\hat{x}_k \in \operatorname{argmin}_{x \in X} L_{\mathcal{S}_k}(x)$ ;
- step 2 (Assess solution quality)**  
 | Calculate  $U_{\mathcal{S}_k}(x)$ . An upper bound on  $\mu_{\hat{x}_k}$  is given by  $U_{\mathcal{S}_k}(\hat{x}_k) - L_{\mathcal{S}_k}(\hat{x}_k)$ ;
- step 3 (Check stopping criterion)**  
 | **if**  $U_{\mathcal{S}_k}(\hat{x}_k) - L_{\mathcal{S}_k}(\hat{x}_k) \leq \varepsilon \cdot \max\{|U_{\mathcal{S}_k}(\hat{x}_k)|, |L_{\mathcal{S}_k}(\hat{x}_k)|\}$  **then** stop;
- step 4 (Improve: Refine partition)**  
 | refine partition  $\mathcal{S}_k$ , let  $k = k + 1$ , update  $v_k$ ; let  $\bar{\xi}^\ell = \mathbb{E}[\xi | \xi \in \Xi_\ell]$  and  
 |  $p^\ell = \mathbb{P}(\xi \in \Xi_\ell), \forall \ell$ ;  
 | goto **step 1**;
- 

Moreover,  $\mathbb{E}[f_k(x, \xi)] \leq \mathbb{E}[f_{k+1}(x, \xi)]$  coupled with (4.10), lower semicontinuity of  $\mathbb{E}[f_k(x, \xi)]$  and continuity of  $c(x) + \mathbb{E}[Q(x, \xi)]$  ensures that  $L_{\mathcal{S}_k}(x) = \mathbb{E}[f_k(x, \xi)]$  converges uniformly to  $c(x) + \mathbb{E}[Q(x, \xi)]$  on compact  $X$  (Kall, 1986, Corollary 7). Under these assumptions  $\lim_{k \rightarrow \infty} L_{\mathcal{S}_k}(\hat{x}_k) = z^*$  and every accumulation point of  $\{\hat{x}_k\}_{k=1}^\infty$  solves (SP2), and existence of such an accumulation point is assured for  $X$  compact.

For these types of results under weaker assumptions via the notion of epi-convergence, see, e.g., Birge and Wets (1986). Frauendorfer (1992) discusses conditions on the refinement schemes to ensure (4.9) holds.

#### 4.4 Sampling-Based Sequential Approximation Method (SSAM)

In this section, we describe an algorithm similar to the SAM of the previous section except that we replace the computationally taxing Edmundson-Madansky upper bound by a sampling-based estimate of  $\mathbb{E}[Q(x, \xi)]$ . The error bounds provided by our new sampling-

based SAM (SSAM) are statistical. This leads us to characterize solution quality in SSAM via a confidence interval rather than the deterministically-valid bounds on the optimality gap of SAM. Moreover, in our sampling-based adaptation of SAM, we iteratively test whether an estimator of the optimality gap satisfies a termination criterion, an inescapably *sequential* procedure. Such sequential procedures can exhibit premature termination with associated poor coverage results for their interval estimators; see, for instance, the discussion by Glynn and Whitt (1992). To avoid premature termination and ensure an asymptotically valid confidence interval on the optimality gap of the solution proposed by SSAM, we adapt the sequential sampling framework of Bayraksan and Morton (2011).

SSAM generates candidate solutions and calculates a lower bound in the same manner as SAM. SSAM also improves this lower bound and hence, the solutions generated, by refining the partition of the random vector's support, again, much like SAM. When forming a sampling-based estimator of  $\mathbb{E}[Q(x, \xi)]$ , the partitioning scheme inherent in SSAM suggests that we employ stratified sampling to reduce the variance of the associated estimator. We attempt to further reduce variance via a control variate, where the control on each cell of the partition are linear approximations of  $Q(x, \cdot)$  that use subgradient information, which is already available. The motivation for using a control variate estimator lies in the notion that the quality of these linear approximations should improve with finer partitions of  $\Xi$ . Indeed, the partitioning scheme already chooses and splits cells for the purpose of reducing nonlinearity of  $Q(x, \cdot)$  to tighten the Jensen bound (see §4.3.3).

We begin our discussion with the stratified sampling-based estimate of the objective function in §4.4.1 and then show how to incorporate the control variate scheme in §4.4.2. To handle the sequential aspect of SSAM, the sampling-based estimates use sample sizes that grow almost linearly with the iteration,  $k$ . We provide rules to increase the sample sizes and to stop SSAM in §4.4.3. We end this section with an algorithmic statement of the SSAM in §4.4.5 and leave the discussion on its theoretical properties to §4.5.

#### 4.4.1 Stratified Monte Carlo Sampling

Given a partition  $\mathcal{S}$  of  $\Xi$  with  $v$  cells, and a solution  $x \in X$ , we estimate  $f(x)$  by stratified Monte Carlo sampling. Suppose we have a sample size of  $N$  to be allocated among the

$v$  cells; i.e.,  $N = \sum_{\ell=1}^v n_\ell$ , where  $n_\ell$  denotes the number of observations drawn from the conditional distribution of  $\xi$  given that  $\xi \in \Xi_\ell$ . We propose a stratified estimator of the form

$$\bar{U}_{\mathcal{J},N}(x) = c(x) + \sum_{\ell=1}^v p^\ell \bar{U}_{n_\ell}^\ell, \quad (4.11)$$

where  $\bar{U}_{n_\ell}^\ell$  estimates  $\mathbb{E}[Q(x, \xi) | \xi \in \Xi_\ell]$  with sample size  $n_\ell$ . For now, we may view  $\bar{U}_{n_\ell}^\ell$  as the standard sample mean estimator, but below we modify it to be a control variate estimator. We prefer a stratified estimator to a crude Monte Carlo estimator because the stratified approach reduces variance, guides selection of cells for refinement (see §4.4.4), and facilitates a control variate to further reduce variance (see §4.4.2).

Ignoring integrality, we allocate the sample size  $N$  across the cells of the partition according to

$$n_\ell = p^\ell N, \quad \ell = 1, 2, \dots, v, \quad (4.12)$$

which ensures variance reduction (Bratley et al., 1987, §2.4). Substituting the sample size formula (4.12) into  $\text{var}[\bar{U}_{\mathcal{J},N}(x)]$  yields

$$\text{var}[\bar{U}_{\mathcal{J},N}(x)] = \frac{1}{N} \sum_{\ell=1}^v p^\ell \sigma_\ell^2(x) \equiv \frac{1}{N} \sigma^2(x), \quad (4.13)$$

and we have the central limit theorem

$$\sqrt{N} (\bar{U}_{\mathcal{J},N}(x) - f(x)) \Rightarrow N(0, \sigma^2(x)), \quad \text{as } N \rightarrow \infty \quad \forall x \in X, \quad (4.14)$$

where  $\Rightarrow$  denotes convergence in distribution. We estimate  $\sigma^2(x)$  using conditional sample variance estimators from each cell,  $s_\ell^2(x)$ , via

$$s^2(x) = \sum_{\ell=1}^v p^\ell s_\ell^2(x). \quad (4.15)$$

In implementation, we round sample sizes up to the next largest integer and require a minimum number of observations per cell.

#### 4.4.2 Control Variates Scheme

We develop a control variate scheme that we employ separately in estimating  $\mathbb{E}[Q(x, \xi) | \xi \in \Xi_\ell]$  at each cell  $\ell$  of the partition. Recall  $\bar{\xi}^\ell = \mathbb{E}[\xi | \xi \in \Xi_\ell]$  and consider

the following first-order Taylor approximation of  $Q(x, \cdot)$  as the control random variable

$$\Gamma_\ell(x, \xi) = Q(x, \bar{\xi}^\ell) + \nabla_\xi Q(x, \bar{\xi}^\ell)(\xi - \bar{\xi}^\ell), \quad \ell = 1, 2, \dots, \nu,$$

where, again,  $\nabla_\xi Q(x, \bar{\xi}^\ell)$  denotes a (sub)gradient of  $Q(x, \cdot)$  at  $\bar{\xi}^\ell$ . When we have  $Q(x, \xi)$  in analytic form, as in the asset allocation problem we use for our computational results in Chapter 5, we can obtain  $\nabla_\xi Q(x, \bar{\xi}^\ell)$  analytically. Or, if  $T(x, \xi) = T(x)$  then the vector of Lagrange multipliers on the constraints of (4.1), say  $\pi(x, \bar{\xi}^\ell)$ , is a subgradient of  $Q(x, \cdot)$  at  $\bar{\xi}^\ell$ . More generally, when  $T(x, \cdot)$  depends on the random parameter, we can use the chain rule to obtain the desired subgradient.

Note that  $\mathbb{E}[\Gamma_\ell(x, \xi) | \xi \in \Xi_\ell] = Q(x, \bar{\xi}^\ell)$  and hence  $\Gamma_\ell(x, \xi)$  has a known conditional mean. SSAM obtains  $Q(x, \bar{\xi}^\ell)$  when solving the lower bounding problem, and the subgradient may also be obtained immediately (e.g., if it has value  $\pi(x, \bar{\xi}^\ell)$ ), or with minimal additional effort (e.g., via the chain rule with an affine  $T(x, \xi)$ ). Therefore, the control,  $\Gamma_\ell(x, \xi)$ , may be formed with essentially no additional computation.

Let

$$W_\ell(x, \xi, \lambda) = Q(x, \xi) - \lambda[\Gamma_\ell(x, \xi) - Q(x, \bar{\xi}^\ell)].$$

We have subtracted from  $Q(x, \xi)$  a random variable with conditional mean zero. Choosing  $\lambda$  to minimize  $\text{var}[W_\ell(x, \xi, \lambda) | \xi \in \Xi_\ell]$  yields

$$\lambda^* = \frac{\text{cov}[Q(x, \xi), \Gamma_\ell(x, \xi) | \xi \in \Xi_\ell]}{\text{var}[\Gamma_\ell(x, \xi) | \xi \in \Xi_\ell]}.$$

We can estimate  $\lambda^*$  by

$$\hat{\lambda}_{n_\ell} = \frac{\widehat{\text{cov}}[Q(x, \xi), \Gamma_\ell(x, \xi) | \xi \in \Xi_\ell]}{\widehat{\text{var}}[\Gamma_\ell(x, \xi) | \xi \in \Xi_\ell]}, \quad (4.16)$$

using the sample covariance and sample variance based on the  $n_\ell$  observations for cell  $\ell$ .

We then form the estimate of  $\mathbb{E}[Q(x, \xi) | \xi \in \Xi_\ell]$ :

$$\bar{W}_{n_\ell}^\ell(\hat{\lambda}_{n_\ell}) = \frac{1}{n_\ell} \sum_{j=1}^{n_\ell} W_\ell(x, \xi^j, \hat{\lambda}_{n_\ell}).$$

Under a joint normality assumption, Lavenberg and Welch (1981) obtain a confidence interval associated with the point estimate,  $\bar{W}_{n_\ell}^\ell(\hat{\lambda}_{n_\ell})$ . Nelson (1990) shows that asymptotically, the normality assumption is unnecessary, obtaining in our notation that as  $n_\ell \rightarrow \infty$

$$\sqrt{n_\ell} \left( \bar{W}_{n_\ell}^\ell(\hat{\lambda}_{n_\ell}) - \mathbb{E}[Q(x, \xi) | \xi \in \Xi_\ell] \right) \Rightarrow N(0, \bar{\sigma}_\ell^2(x)), \quad (4.17)$$

where  $\Rightarrow$  denotes convergence in distribution,  $\bar{\sigma}_\ell^2(x) = (1 - R^2) \sigma_\ell^2(x)$  and  $R^2$  is the square of the conditional correlation coefficient between  $Q(x, \xi)$  and  $\Gamma_\ell(x, \xi)$ .

Instead of estimating  $\lambda^*$ , we can simply use a deterministic value, say,  $\lambda = 1$  (Hammersley and Handscomb, 1964). Then,  $\bar{W}_{n_\ell}^\ell(1)$  is a sample mean of i.i.d. random variables and again satisfies a central limit theorem, albeit with a larger variance term than that of (4.17). In forming our sampling-based estimate of  $f(x)$  we can replace  $\bar{U}_{n_\ell}^\ell$  in (4.11) by  $\bar{W}_{n_\ell}^\ell(\hat{\lambda}_{n_\ell})$  or by  $\bar{W}_{n_\ell}^\ell(1)$ .

The magnitude of variance reduction depends on the correlation between  $Q(x, \xi)$  and  $\Gamma_\ell(x, \xi)$ , where the latter is a linear function of  $\xi$ . As indicated above, the procedure for refining partitions aims to produce cells over which the function  $Q(x, \cdot)$  is nearly linear. Thus, we anticipate that variance reduction should grow as the algorithm converges. We test this effect in our numerical results in Chapter 5, and find favorable results.

#### 4.4.3 Rules to Increase the Sample Sizes and to Stop

To be able to discuss stopping rules, we again start using  $k$  as an index for iteration number. At the  $k$ th iteration of SSAM we use  $G_k$  to estimate the current optimality gap  $\mu_{\hat{x}_k}$ , where

$$G_k = \bar{U}_{\mathcal{J}_k, N_k}(\hat{x}_k) - L_{\mathcal{J}_k}(\hat{x}_k).$$

Extending the notation as in SAM, we use  $N_k$  to denote the total sample size at iteration  $k$ . To simplify the discussion, we take  $\bar{U}_{\mathcal{J}_k, N_k}(\hat{x}_k)$ , the point estimate of  $f(\hat{x}_k)$ , to be the stratified-sampling estimator of equation (4.11) but it could instead be that estimator with  $\bar{U}_{n_\ell}^\ell$  replaced by  $\bar{W}_{n_\ell}^\ell(\hat{\lambda}_{n_\ell})$  or  $\bar{W}_{n_\ell}^\ell(1)$ . Associated with  $G_k$  is a sample variance estimator,  $s_k^2$  from equation (4.15).

Following the sequential sampling framework of Bayraksan and Morton (2011), SSAM stops when  $G_k$  falls below a pre-specified factor of the sample standard deviation,  $s_k$ ; i.e., SSAM terminates at iteration

$$T = \inf_{k \geq 1} \{k : G_k \leq h' s_k + \varepsilon'\}. \quad (4.18)$$

When the procedure stops, a statement on the quality of the candidate solution,  $\hat{x}_T$ , is made as a  $100(1 - \alpha)\%$  approximate confidence interval on its optimality gap of the form

$$[0, hs_T + \varepsilon].$$

Here,  $h > h' > 0$  and  $\varepsilon > \varepsilon' > 0$  are prespecified constants. The terms  $\varepsilon$  and  $\varepsilon'$  are typically small relative to  $h$  and  $h'$ , e.g.  $\varepsilon = 10^{-7}$ , and ensure finite stopping. We discuss how to select  $h$  and  $h'$  in Chapter 5.

Under a finite variance assumption, using a sample size schedule

$$N_k \geq (h - h')^{-2} (c_{p,q} + 2pk^q), \quad (4.19)$$

where  $q > 1$ ,  $p > 0$  and  $c_{p,q} = \max \left\{ 2 \ln \left( \sum_{j=1}^{\infty} \exp[-pj^q] / \sqrt{2\pi\alpha} \right), 1 \right\}$ , it is possible to show that

$$\lim_{h \downarrow h'} \mathbb{P}(\mu_{\hat{x}_T} \leq hs_T + \varepsilon) \geq 1 - \alpha. \quad (4.20)$$

In (4.19),  $q > 1$  and  $p > 0$  are parameters that can be chosen to minimize the computational effort of the sequential sampling procedure. We refer the readers to Bayraksan and Morton (2011) on how to select  $p$  and  $q$ . In §4.5 we will detail the set of conditions to ensure asymptotic validity expressed in (4.20) for SSAM.

#### 4.4.4 Partition Refinement

We described in §4.3.3 how to refine partitions in SAM. That discussion carries over to SSAM directly, except that the error contributed by each cell  $\ell$  to the current estimate of the optimality gap would revise from (4.6) to  $p^\ell [\bar{U}_{n_\ell}^\ell - Q(\hat{x}_k, \bar{\xi}^\ell)]$ . However, we can further revise this by taking into account the stopping criteria (4.18). In particular, we can choose to split cell  $\ell$  if

$$p^\ell \left[ \bar{U}_{n_\ell}^\ell - Q(\hat{x}_k, \bar{\xi}^\ell) - h' s_{k,\ell}(\hat{x}_k) \right] > 0. \quad (4.21)$$

If (4.21) fails to hold for any cell, then the stopping criteria is satisfied because  $\sum_{\ell=1}^v p^\ell s_\ell \leq \left( \sum_{\ell=1}^v p^\ell s_\ell^2 \right)^{1/2}$ . If we find this rule to be too aggressive in splitting cells we can restrict attention to those cells with the largest left-hand side of (4.21). In our implementation, we do this in a matter depicted in Algorithm 2. Our implementation further

uses (4.7) or (4.8) depending on the problem type to select the axis to split, and we split at the conditional mean.

#### 4.4.5 Algorithm Statement

Algorithm 2 details SSAM, beginning with initialization of the parameters,  $h, h', \varepsilon, \varepsilon', \alpha, q$  and  $p$ . The initial partition  $\mathcal{S}_1$  is the entire sample space and so the initial number of cells  $v_1 = 1$ . The list of cells to be considered for splitting, `List`, is initially empty. Therefore, in the first iteration, we just solve the lower-bounding problem in step 1 using Jensen's bound with the single "scenario,"  $\bar{\xi} = \mathbb{E}[\xi]$ . Otherwise, in step 1 of the algorithm, the current partition is refined by selecting cells from `List` to be partitioned. For each cell  $\Xi_\ell$  with  $\ell \in \text{List}$  we split  $\Xi_\ell$  by selecting the axis according to (4.7) or (4.8) depending on the problem type and splitting at the conditional mean as indicated in §4.4.4.

Step 2 chooses a sample size according to (4.19), allocates samples to each cell proportional to the probability mass of that cell, and forms the conditional sample mean and conditional sample variance. In each cell of the partition, we form these estimators by generating i.i.d. observations from the distribution of  $\xi$ , conditional on  $\xi$  being in that cell. These samples are also independent of samples generated in previous iterations. We then form a point estimate of the objective function value for the current candidate solution under stratified sampling according to (4.11) and the associated sample variance estimator according to (4.15). After forming the point estimate of the optimality gap,  $G_k$ , we check the stopping criterion in step 3.

While we describe the algorithm in step 2 for the simplest case of stratified sampling, it is straightforward to modify step 2 in an attempt to further reduce variance using the control variate scheme described in §4.4.2. Step 4 of the algorithm selects cells of the current partition to consider for refinement. Instead of selecting all cells according to condition (4.21), it selects the cells that contribute at least  $100r\%$  to the current violation. We select  $r$  to find a balance between the number of cells in the partition and the convergence of the algorithm; we further discuss this in our computational implementations in Chapter 5.

---

**Algorithm 2: Sampling-based Sequential Approximation Method (SSAM)**


---

**step 0 (Initialization)**

select  $h > h' > 0$ ,  $\varepsilon > \varepsilon' > 0$ ,  $0 < \alpha < 1$ ,  $q > 1$  and  $p > 0$  ;  
 let  $k = 1$ ,  $\Xi_k = \Xi$ ,  $\mathcal{S}_k = \{\Xi_k\}$ ,  $\text{List} = \emptyset$ ,  $v_k = 1$ ,  $LB_k = -\infty$ ;

**step 1 (Generate candidate solution)**

$\hat{x}_k \in \operatorname{argmin}_{x \in X} L_{\mathcal{S}_k}(x)$ ,  $v_k = |\mathcal{S}_k|$ ,  $LB_k = L_{\mathcal{S}_k}(\hat{x}_k)$ ;

**step 2 (Assess solution quality)**

**for**  $\ell = 1$  **to**  $v_k$  **do**  
 | allocate  $n_\ell = \lceil p^\ell N_k \rceil$  samples to cell  $\ell$ ; form  $\bar{U}_{n_\ell}^\ell$  and  $s_{k,\ell}^2(\hat{x}_k)$ ;  
**end**  
 $\bar{U}_{\mathcal{S}_k, N_k}(\hat{x}_k) = c(\hat{x}_k) + \sum_{\ell=1}^{v_k} p^\ell \bar{U}_{n_\ell}^\ell$ ,  $s_k^2 = \sum_{\ell=1}^{v_k} p^\ell s_{k,\ell}^2(\hat{x}_k)$ ,  
 $G_k = \bar{U}_{\mathcal{S}_k, N_k}(\hat{x}_k) - L_{\mathcal{S}_k}(\hat{x}_k)$ ;

**step 3 (Check stopping criterion)**

**if**  $G_k \leq h's_k + \varepsilon'$  **then**  
 | output  $\hat{x}_k$  and confidence interval  $[0, h's_k + \varepsilon]$  on  $\mu_{\hat{x}_k}$  and stop;  
**end**

**step 4 (Improve: Refine partition and increase sample size)**

Let  $0 \leq r < 1$ ,  $\Delta_k = G_k - h's_k$ ,  $\delta_k^\ell = p^\ell \left( \bar{U}_{n_\ell}^\ell - Q(\hat{x}_k, \bar{\xi}^\ell) - h's_k^\ell \right)$ ;  
 $\text{List} = \left\{ \ell : \sum_{\ell \in \text{List}} \delta_k^\ell > r\Delta_k, \text{ and } \delta_k^\ell \geq \delta_k^{\ell_1} \ \forall \ell_1 \notin \text{List} \right\}$  and  $|\text{List}|$  is  
 minimal;  
 Set  $k = k + 1$ ;  
**for**  $\ell \in \text{List}$  **do**  
 | split  $\Xi_\ell$  and update  $\mathcal{S}_k$ ; compute  $p^{\ell'}$  and  $\bar{\xi}^{\ell'}$  for two new cells;  
**end**  
 Set  $N_k = \left\lceil (h - h')^{-2} (c_{p,q} + 2pk^q) \right\rceil$  and goto **step 1**;

---

#### 4.5 Asymptotic Validity

In this section, we establish the asymptotic validity of the confidence interval output by the SSAM as outlined in Algorithm 2 and show that the procedure stops in a finite number of iterations with probability one. By nature of the statistical upper bound and the stopping rule used in SSAM, the iteration number  $T$  when the procedure stops and the candidate solution output  $\hat{x}_T$  are random variables and their analysis must be probabilistic. Because SSAM uses the stopping rule provided by Bayraksan and Morton (2011), proofs of asymptotic validity and finite stopping follow in the same spirit as those for the sequential sampling procedure developed by Bayraksan and Morton (2011), but with unique features for SSAM.

Given a sample size  $N$  and a partition  $\mathcal{S}$ , let

$$D_{\mathcal{S},N}(x) = \bar{U}_{\mathcal{S},N}(x) - f(x^*). \quad (4.22)$$

Here, for simplicity, we view  $\bar{U}_{\mathcal{S},N}(x)$  as the stratified sample mean estimator described in §4.4.1 and go through the proof with this estimator. Notice that in this case  $\mathbb{E}[D_{\mathcal{S},N}(x)] = \mu_x$  and  $\text{var}[D_{\mathcal{S},N}(x)] = N^{-1}\sigma^2(x)$ , as given in (4.13), along with its sample variance estimator,  $s^2(x)$ , given in (4.15).

One of the main differences in the proof of asymptotic validity of SSAM compared to the general relative-width stopping rules theory provided by Bayraksan and Morton (2011) is the establishment of a required uniform boundedness condition. We will do this for SSAM below in Lemma 7. Before we begin, we simplify our notation. First, at iteration  $k$  of SSAM, we use  $N_k$  samples within partition  $\mathcal{S}_k$ , and the candidate solution is denoted by  $\hat{x}_k$ . We suppress the dependence on these and only use iteration  $k$ . For instance, we denote  $D_{\mathcal{S}_k, N_k}(\hat{x}_k)$  simply as  $D_k$  and  $\mu_{\hat{x}_k}$  as  $\mu_k$ . Notice that  $D_k \geq G_k$  w.p.1 for all  $k \geq 1$ , i.e.,  $G_k = \bar{U}_{\mathcal{S}_k, N_k}(x) - L_{\mathcal{S}_k}(\hat{x}_k) \geq \bar{U}_{\mathcal{S}_k, N_k}(x) - f(x^*) = D_k$ , because  $L_{\mathcal{S}}(x) \leq f(x^*)$  for all  $x \in X$  and partition  $\mathcal{S}$ . We also let  $\mathcal{H}_k = (\hat{x}_k, \mathcal{S}_k)$  denote the current partition  $\mathcal{S}_k$  along with the candidate solution  $\hat{x}_k$  obtained by solving  $\min_{x \in X} L_{\mathcal{S}_k}(x)$ . In the lemma below, we stipulate how fast the sample sizes  $N_k$  must grow in order to ensure a uniform boundedness condition.

**Lemma 7.** *Assume (A10) holds. Let  $\varepsilon > \varepsilon' > 0$  and  $h > h' > 0$  be fixed and let  $\Delta\varepsilon = \varepsilon - \varepsilon'$  and  $\Delta h = h - h'$ . Consider SSAM where  $N_k$  is allocated as in (4.12) to the cells in  $\mathcal{S}_k$ . If  $N_k$  grows proportional to  $k^{1+\delta}$  for some  $\delta > 0$ , then  $\sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq \Delta h s_k - \Delta\varepsilon) < \infty$ .*

*Proof.* Let  $m_k = \mathbb{E}[Q(\hat{x}_k, \xi)]$  and  $\sigma_k^2 = \text{var}[Q(\hat{x}_k, \xi)]$ . Conditional means and variances at iteration  $k$  are similarly denoted as  $m_k^\ell = \mathbb{E}[Q(\hat{x}_k, \xi) | \xi \in \Xi_\ell]$  and  $\sigma_{k,\ell}^2 = \text{var}[Q(\hat{x}_k, \xi) | \xi \in \Xi_\ell]$  for  $\ell = 1, 2, \dots, v_k$ . Also, let  $\bar{Q}_k^\ell = \frac{1}{n_\ell} Q(\hat{x}_k, \xi^i)$ , where the  $\xi^i$ 's form a random sample from the distribution of  $\xi$  over  $\Xi_\ell$ ,  $\ell = 1, 2, \dots, v_k$ .

$$\begin{aligned}
\sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta h s_k - \Delta\varepsilon) &\leq \sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta\varepsilon) \\
&\leq \sum_{k=1}^{\infty} \mathbb{P}(|D_k - \mu_k| \geq \Delta\varepsilon) \\
&= \sum_{k=1}^{\infty} \int_{\mathcal{H}_k} \mathbb{P}(|D_k - \mu_k| \geq \Delta\varepsilon | \mathcal{H}_k) d\mathbb{P}_{\mathcal{H}_k} \\
&= \sum_{k=1}^{\infty} \int_{\mathcal{H}_k} \mathbb{P}((D_k - \mu_k)^2 \geq (\Delta\varepsilon)^2 | \mathcal{H}_k) d\mathbb{P}_{\mathcal{H}_k} \\
&\leq \sum_{k=1}^{\infty} \int_{\mathcal{H}_k} \mathbb{E}[(D_k - \mu_k)^2 | \mathcal{H}_k] (\Delta\varepsilon)^{-2} d\mathbb{P}_{\mathcal{H}_k} \quad (4.23)
\end{aligned}$$

where (4.23) follows from an application of Markov's inequality. Note that  $D_k - \mu_k = \left(\sum_{\ell=1}^{v_k} p^\ell \bar{Q}_k^\ell\right) - m_k = \sum_{\ell=1}^{v_k} p^\ell (\bar{Q}_k^\ell - m_k^\ell)$ . Now,

$$\begin{aligned}
\mathbb{E} \left[ \left( \sum_{k=1}^{v_k} p^\ell (\bar{Q}_k^\ell - m_k^\ell) \right)^2 \middle| \mathcal{H}_k \right] &= \sum_{\ell=1}^{v_k} (p^\ell)^2 \mathbb{E} \left[ (\bar{Q}_k^\ell - m_k^\ell)^2 \middle| \mathcal{H}_k \right] \\
&= \sum_{\ell=1}^{v_k} (p^\ell)^2 \frac{\sigma_{k,\ell}^2}{n_\ell} \\
&= \frac{1}{N_k} \sum_{\ell=1}^{v_k} (p^\ell \sigma_{k,\ell}^2) \quad (4.24)
\end{aligned}$$

$$\leq \frac{1}{N_k} \sigma_k^2, \quad (4.25)$$

where (4.24) follows from the proportional allocation rule,  $n_\ell = p^\ell N_k$  and the inequality (4.25) follows from the variance decomposition formula (i.e., the law of total variation):

$$\text{var}[Q(x, \xi)] = \mathbb{E}[\text{var}[Q(\hat{x}_k, \xi) | \xi \in \Xi_\ell]] + \text{var}[\mathbb{E}[Q(\hat{x}_k, \xi) | \xi \in \Xi_\ell]].$$

Combining (4.23) and (4.25), and taking the supremum of variances we obtain

$$\begin{aligned} \sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta h s_k - \Delta \varepsilon) &\leq \frac{\sup_{x \in X} \text{var}[Q(x, \xi)]}{(\Delta \varepsilon)^2} \sum_1^k \frac{1}{N_k} \\ &< \infty, \end{aligned}$$

where the last inequality follows from the fact that  $N_k \propto k^{1+\delta}$  for some  $\delta > 0$  and  $\sup_{x \in X} \text{var}[Q(x, \xi)] < \infty$  by (A10).  $\square$

Next, we show the asymptotic validity of SSAM. To be able to do this we need to use Fatou's Lemma and a bound on the tail of a standard normal, which are provided in the next two lemmas.

**Lemma 8** (Fatou's Lemma). *Suppose  $\{f_n\}$  is a sequence of measurable functions on  $E$ .*

(i) *If  $f_n \geq 0$  for all  $n$ , then  $\int_E \liminf_{n \rightarrow \infty} f_n \leq \liminf_{n \rightarrow \infty} \int_E f_n$ .*

(ii) *If  $L \leq f_n \leq U$  for all  $n$ , such that  $\int_E L < \infty$  and  $\int_E U < \infty$ , then*

$$\int_E \liminf_{n \rightarrow \infty} f_n \leq \liminf_{n \rightarrow \infty} \int_E f_n \leq \limsup_{n \rightarrow \infty} \int_E f_n \leq \int_E \limsup_{n \rightarrow \infty} f_n.$$

**Lemma 9** (Bound on Tail of a Standard Normal). *Let  $Z$  be a standard normal and  $t > 0$ .*

*Then,*

$$\mathbb{P}(Z \geq t) \leq \frac{1}{\sqrt{2\pi}} \frac{\exp(-t^2/2)}{t}.$$

**Theorem 10.** *Assume (A8)-(A10) hold. Let  $\varepsilon > \varepsilon' > 0, p > 0, q > 1$  and  $0 < \alpha < 1$  be fixed. Then, for SSAM where the sample sizes are increased according to (4.19) and allocated according to (4.12) to the cells in  $\mathcal{S}_k$ , and the procedure stops at iteration  $T$  according to (4.18),*

$$\liminf_{h \downarrow h'} \mathbb{P}(\mu_T \leq h s_T + \varepsilon) \geq 1 - \alpha.$$

*Proof.* Let  $\Delta h = h - h'$  and  $\Delta \varepsilon = \varepsilon - \varepsilon'$ . Then,

$$\begin{aligned}
\mathbb{P}(\mu_T > hs_T + \varepsilon) &\leq \mathbb{P}(\mu_T > G_T + \Delta hs_T + \Delta \varepsilon) & (4.26) \\
&= \sum_{k=1}^{\infty} \mathbb{P}(G_1 > h's_1 + \varepsilon', \dots, G_{k-1} > h's_{k-1} + \varepsilon', \\
&\quad G_k \leq h's_k + \varepsilon', G_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon) \\
&\leq \sum_{k=1}^{\infty} \mathbb{P}(G_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon) \\
&\leq \sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon), & (4.27)
\end{aligned}$$

where (4.26) follows from (4.18) and (4.27) follows from the fact that  $G_k \geq D_k$ , w.p.1 for all  $k \geq 1$ . So, it suffices to show that  $\lim_{\Delta h \downarrow 0} \sup \sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon) \leq \alpha$ . We will apply part (ii) of Fatou's Lemma to continue the proof. This requires the right-hand side of (4.27) to be bounded and Lemma 7 guarantees this with  $N_k$  given in (4.19).

Taking limits, we obtain

$$\begin{aligned}
&\limsup_{\Delta h \downarrow 0} \sum_{k=1}^{\infty} \mathbb{P}(D_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon) \\
&\leq \sum_{k=1}^{\infty} \limsup_{\Delta h \downarrow 0} \mathbb{P}(D_k - \mu_k \leq -\Delta hs_k - \Delta \varepsilon) \\
&\leq \sum_{k=1}^{\infty} \limsup_{\Delta h \downarrow 0} \int_{\mathcal{H}_k} \mathbb{P}\left(\frac{D_k - \mu_k}{\sigma_k / \sqrt{N_k}} \leq -\Delta h \sqrt{N_k} \left(\frac{s_k}{\sigma_k}\right) \middle| \mathcal{H}_k\right) d\mathbb{P}_{\mathcal{H}_k} \\
&\leq \sum_{k=1}^{\infty} \int_{\mathcal{H}_k} \limsup_{\Delta h \downarrow 0} \mathbb{P}\left(\frac{D_k - \mu_k}{\sigma_k / \sqrt{N_k}} \leq -\sqrt{c_{p,q} + 2pk^q} \left(\frac{s_k}{\sigma_k}\right) \middle| \mathcal{H}_k\right) d\mathbb{P}_{\mathcal{H}_k} \\
&\leq \alpha,
\end{aligned}$$

where the first and third inequalities follow from Fatou's Lemma. With  $\mathcal{H}_k = (\hat{x}_k, \mathcal{S}_k)$  fixed, we have a stratified estimator over partition  $\mathcal{S}_k$  and the central limit theorem holds as in (4.14) since  $\Delta h \downarrow 0$  ensures that  $N_k \rightarrow \infty$  and  $n_\ell \rightarrow \infty$  whenever  $p^\ell > 0$ . Similarly,  $\lim_{\Delta h \downarrow 0} (s_k / \sigma_k) = 1$ , w.p.1 with  $\mathcal{H}_k$  fixed for the stratified estimator. The last inequality follows from Lemma 9 and applying the definition of  $c_{p,q}$ .  $\square$

Above we assumed the use of the stratified estimator  $\bar{U}_{n_\ell}^\ell$ . We can easily change this to  $\bar{W}_{n_\ell}(1)$  as it satisfies the central limit theorem with i.i.d. sampling. We can also use

$\bar{W}_{n_\ell}(\hat{\lambda}_{n_\ell})$  with further assumptions but we do not pursue it here. We also note that finite stopping of the algorithm, i.e.,  $\mathbb{P}(T < \infty) = 1$ , can be shown under conditions that ensure that accumulation points of  $\{\hat{x}_k\}_{k=1}^\infty$  solves (SP2), w.p.1 (see §4.3.4 for discussion on this) and for any subsequence of  $\{\hat{x}_k\}_{k=1}^\infty$  that converge to  $x \in X$ ,  $\liminf_{k \rightarrow \infty} \mathbb{P}(|G_k - \mu_x| > \delta) = 0$  for any  $\delta > 0$ . See proof of Proposition 1 in Bayraksan and Morton (2011) for details.

#### 4.6 Summary and Concluding Remarks

In this chapter, we developed an algorithm for solving two-stage stochastic programming with a convex second stage program and with uncertainty in the right-hand side. The algorithm is inspired from both sampling and bounding techniques, derived from sequential sampling and classical sequential approximation methods. The support of the random vector is partitioned and the sample size is increased as the algorithm progresses. Rules for partitioning, increasing the sample sizes, and for allocating the sample sizes among the cells in the partition have been developed. Key theoretical results, such as asymptotic validity of the confidence interval output, have been established. In the next chapter, we examine the performance of these procedures on test problems from the literature.

## CHAPTER 5

## SSAM: COMPUTATIONAL RESULTS

In this chapter, we report on computational experiments with the SSAM. We first study the effectiveness of the proposed variance reduction techniques by experimenting on an asset allocation problem from the literature. We then report our computational experiments on this asset allocation problem as well as different dimensional instances of the 20TERM model. This chapter is organized as follows. In §5.1, we describe the test problems used for the numerical results reported. Setting of the values for the various parameters of the SSAM algorithm is discussed in §5.2. We study the variance reduction in SSAM in §5.3. In §5.4, we present the empirical performance of SSAM on an asset allocation model from the literature. In §5.5, we study how the algorithm fares as the dimension of the random vector increases. We end with a summary and concluding remarks in §5.6.

## 5.1 Test Problems

We consider instances from two test problems for our computational experiments. The first one, an asset allocation problem, is a two-stage stochastic convex program whereas the second one is a two-stage stochastic linear program with recourse. We experimented on an asset allocation problem to study the extent of variance reduction in SSAM as well as the empirical performances of the algorithm. The asset allocation problem with  $m$  assets is defined as

$$z^* = \max_{x \in X} \mathbb{E}[u(x, \xi)], \quad (5.1)$$

where  $X = \{x : \sum_{j=1}^m x_j = 1, x_j \geq 0, j = 1, \dots, m\}$ ,  $u(x, \xi) = 1 - e^{-\rho(\xi x)}$  and  $\xi x = \sum_{j=1}^m \xi_j x_j$ . We use problem data for model (5.1) from (Partani et al., 2006), which specifies the mean vector and covariance matrix for a 14-asset model in which the random return vector  $\xi$  follows a multivariate normal distribution. We depart from Partani et al. (2006) in that we use an exponential, rather than a power, utility function, but we use a

risk parameter of  $\rho = 0.812211$  in an attempt to match their optimal solution. Our optimal solution,  $x^*$ , has three non-zero components ( $x_4 = 0.0786$ ,  $x_9 = 0.7317$ , and  $x_{10} = 0.1897$ , using the same ordering as Partani et al. (2006)) and  $z^* = 0.5881$ . All problem instances of (5.1) were solved using CVX version 1.21 (Grant and Boyd, 2008, 2011) in MATLAB on a 2.93GHz Dell Xeon multi-processor computer with 9.0 GB of memory, using only one processor. We used the pre-defined MATLAB utilities to generate the normal random variates.

To study the performance of SSAM as the dimension of the random vector increases, we generated six new instances of a motor freight scheduling problem, which is referred to as 20TERM Mak et al. (1999), a two-stage stochastic linear program with recourse. 20TERM is described in Example 1.3. We repeat this description for convenience. Recall that the 20TERM model corresponds to a motor freight carrier with a fleet of vehicles (e.g., tractors and trailers) to carry shipments (customer demands) to-and-fro a central terminal, local terminals, and other customer shipment centers. The operations of the motor freight carrier are conducted as follows. First, the manager of the carrier decides at which terminals to position a fleet of vehicles at the beginning of the day while customer shipment demands are unknown. Then once demands are observed, decisions need to be made to move the fleet through a multi-commodity network in order to satisfy shipment demands in such a way that all the vehicles return back to their initial positions at the end of the day. Description of this model is provided by Mak et al. (1999). We generated different instances of 20TERM by varying the number of random demand points from the original model data used in Mak et al. (1999). We provide further details on how these instances were created in §5.5. Instances of 20TERM were all solved using the regularized decomposition algorithm of Ruszczyński (1986) and Ruszczyński and Świetanowski (1997) on a 2.93GHz Dell Xeon multi-processor computer with 9.0 GB of memory, using only one processor.

## 5.2 Parameter Settings

The values of the parameters used in the algorithm are set as follows. We set  $\alpha = 0.1$  so that the procedure outputs a 90% confidence interval on the optimality gap of the candidate solution. We set  $q = 1.5$  and  $p = 4.67 \times 10^{-3}$  according to the guidelines in Bayraksan and Morton (2011), and we determine  $\Delta h = h - h'$  so that the sample size at the first iteration  $k = 1$  is at least as large as some initial target sample size,  $N_0$ . For instance, we can let  $\Delta h = 0.3114$  for a targeted initial sample size  $N_1 = 100$ . Choosing values for  $h'$ , however, requires more care. If  $h'$  is chosen too large compared to the values of the ratio  $G_T/s_T$ , then the procedure may stop too quickly and yield too large a confidence interval on the optimality gap; whereas if  $h'$  is selected too small then the procedure might run for too long with an excessive number of iterations. In selecting specific values for  $h$  and  $h'$ , we thus proceed as follows. For each test problem and with a moderate initial sample size, we examine the average values of  $G_k/s_k$  for the first few iterations (typically for  $k \leq 15$ ) over 25 replications. We can then select  $h'$  to be slightly smaller than the average of values of the  $G_k/s_k$  ratio observed during the pilot runs. The value of  $h$  is then set such that  $h = h' + \Delta h$ . As the average values of  $G_k/s_k$  vary with the test problem and with the use of control variates, values for  $h'$  will be dependent on these factors as well. We let  $\varepsilon = 2 \times 10^{-8}$  and  $\varepsilon' = 1 \times 10^{-8}$ . In implementation, these two parameters are used for the purpose of numerical tolerances.

## 5.3 Variance Reduction

We study variance reduction in SSAM for the different estimators we have proposed. Our simplest estimator, a stratified estimator, allocates samples proportional to each cell's probability mass according to equation (4.12). This stratified estimator with proportional sampling (STP) already reduces variance, and to assess this reduction we also employ a crude Monte Carlo (CMC) estimator. First, by the definition of the sample variance, we expect the variance to decrease as the sample size grows. Thus, we should gain some variance reduction as the initial sample size  $N_0$  selected for the procedure increases. Second, we can exploit the stratified sampling approach to induce variance reduction. When

the sample size allocation among the cells is done proportionally to the probability of each cell  $l$  in the current partition  $\mathcal{S}_k$  (i.e.,  $n_\ell = p^\ell N_k$  for  $\ell = 1, 2, \dots, v_k$ ), then we are guaranteed sample estimates from distributions with smaller variances. We can use the control variate estimator of §4.4.2 within the stratified sampling scheme in an attempt to further reduce variance. This estimator has a parameter  $\lambda$ , and we can either fix  $\lambda = 1$  (SCV1) or we can estimate  $\lambda$  ( $\widehat{\text{SCV}}$ ) by the  $\hat{\lambda}_{n_\ell}$  of equation (4.16).

We compare the sample variances of these four estimators as follows. For  $h' = 0.5275$ , varying initial sample sizes from  $N_0 = 100$  up to  $N_0 = 500$ , we made 100 runs of SSAM for each  $N_0$  in which the sampling-based upper bound estimator uses the  $\widehat{\text{SCV}}$  procedure. For each run, we compute the other estimators (CMC, STP and SCV1) in the background using the same sample sizes, as the algorithm progresses. For each estimator, we take the average of the variance estimates at iteration  $k$ ,  $s_k^2$ , over the 100 runs. Table 5.1 compares the estimated variances between each pair of these sampling schemes by computing the ratio of the average of these variances as the procedure progresses for the first 10 iterations when  $N_0 = 100$ .

$k$	CMC/ $\widehat{\text{SCV}}$	STP/ $\widehat{\text{SCV}}$	SCV1/ $\widehat{\text{SCV}}$	CMC/SCV1	STP/SCV1	CMC/STP
1	11.60	12.26	1.15	10.11	10.69	0.95
2	84.67	35.64	1.82	46.45	19.55	2.38
3	243.67	65.04	1.83	133.13	35.53	3.75
4	479.57	98.77	1.51	318.38	65.57	4.86
5	640.33	124.27	2.45	261.73	50.79	5.15
6	750.18	136.27	2.39	314.19	57.07	5.51
7	1090.54	175.60	1.99	548.66	88.35	6.21
8	1136.75	169.23	2.06	550.65	81.98	6.72
9	1415.96	197.21	2.42	584.34	81.39	7.18
10	1575.84	205.94	2.65	593.56	77.57	7.65

Table 5.1: Ratio of variance estimators for each pair of sampling scheme.

According to the results in Table 5.1,  $\widehat{\text{SCV}}$  dominates the other estimators. The second column indicates that  $\widehat{\text{SCV}}$  reduces variance by a factor ranging from 11 to 1575 over crude Monte Carlo sampling, the third column indicates  $\widehat{\text{SCV}}$  improves over the simple stratified estimator by factors ranging roughly from 12 to 205, and the fourth column indicates that estimating  $\lambda$  improves variance ratios by a factor of 1 to 2. Table 5.1 suggests the following ordering in terms of variance reduction over CMC:  $\widehat{\text{SCV}} > \text{SCV1} > \text{STP} >$

CMC. Similar results are obtained for larger initial sample size values  $N_0 = 200, 300, 400,$  and  $500$ . Since the sampling scheme  $\widehat{SCV}$  results in the largest variance reduction, we use that estimator to form the sampling-based upper bound in SSAM in the remainder of the results we report in this section.

#### 5.4 Empirical Results on the Asset Allocation Model

In this section, we report on the performance of the SSAM on the asset allocation problem instance described in §5.1. A common random number stream is used in order to eliminate variation due to sampling in the performance of the algorithm across runs.

Table 5.2 summarizes empirical performance measures of SSAM for initial sample size values of  $N_0 = 1000, 2000, 3000, 4000,$  and  $5000$ . For each value of the initial sample size, we report 90% confidence intervals on the following: stopping iteration  $T$ ; the sample size used at termination  $N_T$ ; the number of scenarios (number of cells in the partition) used for the lower-bounding problem at termination;  $hs_T + \varepsilon$ , the width of the confidence interval on the optimality gap; and, the average run time based on the results of 100 independent runs. The last two columns of Table 5.2 report, respectively from left to right, the proportion of  $z^*$  represented by the average of the width of the 90% CI on the optimality gap  $[0, hs_T + \varepsilon]$  and the corresponding true average relative optimality gap (true optimality gap divided by  $z^*$ ). Overall, the algorithm performs well on our asset allocation model. The empirical coverage probability is 100%, the procedure requires a modest number of iterations to stop (on average  $T < 6$ ), and the number of cells in the partition is relatively small. However, the confidence intervals on the optimality gap appear to be somewhat conservative. For instance, even though the CI width on the optimality gap ranges from 1.35% to 1.14% of the optimal value, more than 80% of the solutions,  $\hat{x}_T$ , that the procedure produced are actually within 0.5% of optimality and the averages of the true optimality gaps of the solutions obtained by SSAM are all within 0.12% of  $z^*$ .

$N_0$	$h$	$T$	$N_T$	# Cells	$hs_T + \varepsilon (\times 10^{-3})$	Time(s)	% Opt	True % Opt
1000	0.6260	5.66 $\pm$ 0.74	1017.42 $\pm$ 4.19	5.95 $\pm$ 0.98	7.9 $\pm$ 1.1	13.49	1.35	0.11
2000	0.5971	4.91 $\pm$ 0.52	2027.04 $\pm$ 5.00	5.02 $\pm$ 0.62	7.5 $\pm$ 0.9	18.53	1.28	0.10
3000	0.5844	4.46 $\pm$ 0.29	3023.94 $\pm$ 3.19	4.46 $\pm$ 0.29	7.0 $\pm$ 0.6	19.93	1.19	0.10
4000	0.5767	4.47 $\pm$ 0.36	4044.33 $\pm$ 5.76	4.47 $\pm$ 0.36	7.2 $\pm$ 0.6	23.47	1.23	0.12
5000	0.5715	4.21 $\pm$ 0.21	5050.10 $\pm$ 3.55	4.21 $\pm$ 0.21	6.7 $\pm$ 0.5	24.18	1.14	0.09

Table 5.2: Empirical results on the asset allocation problem, where  $h' = 0.5275$ .

### 5.5 Performance of SSAM as $d_\xi$ increases

In this section, we examine how the performance of SSAM varies with the dimension of the random vector. To this end, we created various instances of the test problem 20TERM by fixing some selected components of the random vector to their expected values. Recall that the original 20TERM model (Mak et al., 1999) corresponds to a freight transportation problem where demands to be satisfied and shipped in both directions between consolidation centers (CCs) and local terminals (LTs) are random. This model is built with the convention that if a demand from a given CC to a given LT (symbolically,  $CC \rightarrow LT$ , where the arrow denotes the direction of shipment) is random, so is the demand in the reverse direction (i.e., from that given LT to the CC or symbolically  $CC \leftarrow LT$ ). With that convention, there are 20  $CC \rightarrow LT$  shipments with random demands (hence the name 20TERM for the model) which results in a total of 40 individual random demands. We generate instances of 20TERM with a smaller number of random parameters by modifying the data from the original model as follows. To generate instances with an even number of random parameters (such as 10, 20, 30, and 36), we set a fixed number of pairs of  $CC \rightarrow LT$  and  $CC \leftarrow LT$  demands to their expected values. For example, to obtain a problem instance with 30 random demand parameters we can keep stochastic every fourth  $CC \rightarrow LT$  demand (from the original 20) as well as their counterparts  $CC \leftarrow LT$ . Proceeding this way, we also manage to preserve the symmetry of the original 20TERM model. To generate instances with an odd number of random parameters, such as instances with 15 and 25 random demands, we treat the demands  $CC \rightarrow LT$  and  $CC \leftarrow LT$  separately. For example, when we fix every second  $CC \rightarrow LT$  demand to their expected values while keeping stochastic every fourth  $CC \leftarrow LT$  demand, we obtain problem instances with 15 random demands. Proceeding this way, we created six new problem instances derived from 20TERM plus the

original 20TERM problem instance with the following dimensions of the random vector: 10, 15, 20, 25, 30, 36, and 40. Characteristics of these instances are summarized in Table 5.3. We label each instance so as to reflect the number of random demands from each type of  $CC \rightarrow LT$  and  $CC \leftarrow LT$  shipments. For example, the instance  $n-m$ :TERM has  $n$  random demands for shipments of the type  $CC \rightarrow LT$  and  $m$  random demands for shipments of the type  $CC \leftarrow LT$ . Each random demand has 2 realizations. With this notation,  $20-20$ :TERM corresponds to the original 20TERM model.

Instance name	# of stochastic parameters	# of 1 <sup>st</sup> stage variables	# of 2 <sup>nd</sup> stage variables	# of scenarios
5-5:TERM	10	63	$7.82 \times 10^5$	$1.02 \times 10^3$
10-5:TERM	15	63	$2.50 \times 10^7$	$3.28 \times 10^4$
10-10:TERM	20	63	$8.01 \times 10^8$	$1.05 \times 10^6$
15-10:TERM	25	63	$2.56 \times 10^{10}$	$3.36 \times 10^7$
15-15:TERM	30	63	$8.20 \times 10^{11}$	$1.07 \times 10^9$
18-18:TERM	36	63	$5.25 \times 10^{13}$	$6.87 \times 10^{10}$
20-20:TERM	40	63	$8.40 \times 10^{14}$	$1.10 \times 10^{12}$

Table 5.3: Characteristics of the derived instances from the 20TERM model

Instance	T	$N_T$	# Cells	$hs_T + \epsilon$	Time(s)	% Opt
5-5:TERM	9.10 $\pm$ 0.17	1032.68 $\pm$ 0.86	10.06 $\pm$ 0.28	2273.88 $\pm$ 106.04	126.35	0.93
10-5:TERM	9.82 $\pm$ 0.52	1035.01 $\pm$ 2.60	11.30 $\pm$ 0.72	3259.89 $\pm$ 116.17	173.78	1.34
10-10:TERM	21.28 $\pm$ 0.46	1126.72 $\pm$ 4.95	48.35 $\pm$ 1.68	3761.63 $\pm$ 79.45	539.76	1.53
15-10:TERM	25.05 $\pm$ 0.54	1179.31 $\pm$ 8.38	61.71 $\pm$ 2.11	4064.30 $\pm$ 65.32	782.24	1.65
15-15:TERM	29.97 $\pm$ 0.51	1385.87 $\pm$ 25.06	124.09 $\pm$ 4.11	4989.32 $\pm$ 85.62	1324.56	2.01
18-18:TERM	35.10 $\pm$ 0.34	2714.49 $\pm$ 86.71	271.35 $\pm$ 8.69	5342.75 $\pm$ 61.31	2659.05	2.14
20-20:TERM	40.02 $\pm$ 0.33	4601.10 $\pm$ 126.84	460.11 $\pm$ 12.68	5674.84 $\pm$ 66.58	4343.62	2.27

Table 5.4: Performance of SSAM as  $d_\xi$  increases.  $N_0 = 1000$ ,  $h' = 1.500$ .

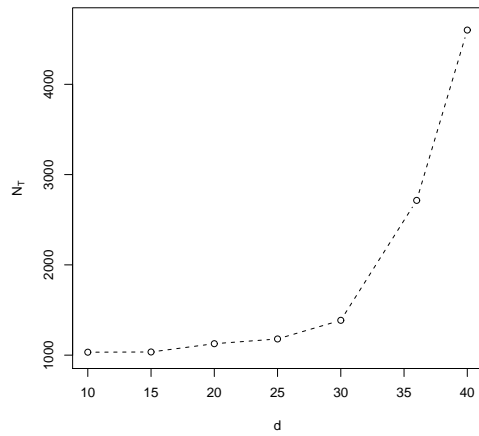
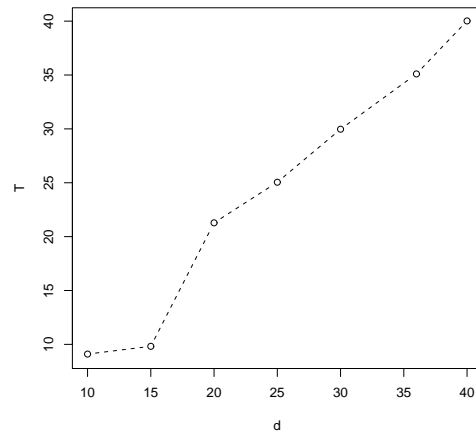
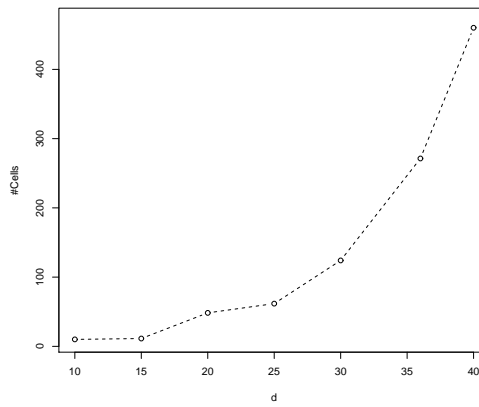
In Table 5.4, we report the empirical performance of SSAM on the seven problem instances described in Table 5.3. We study the performance of SSAM as the dimension  $d_\xi$  increases by comparing the empirical performance results across the seven instances of 20TERM over 100 independent runs for an initial sample size  $N_0 = 1000$  and  $h = 1.5985$ ,  $h' = 1.5000$ . The performance metrics of Table 5.4 are computed in the same fashion as those of Table 5.2 except for the last column. For these results, as  $z^*$  is unknown for all these 20TERM instances but one, we estimated  $z^*$  of each instance with the largest lower

bound value ( $z_{LB}^{\max}$ ) generated by SSAM over the execution of the 100 runs. We then use  $z_{LB}^{\max}$  instead of  $z^*$  in calculating the numbers reported in column % Opt.

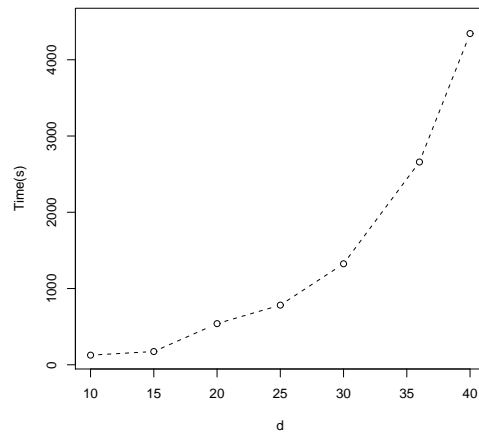
To help better understand the trends of the different performance measures reported, we provide in Figure 5.1 plots of the averages of the performance measures reported in Table 5.4. The plot in Figure 5.1(b) shows an approximately linear growth for the stopping iteration with respect to the dimension of the random vector. According to Figures 5.1(a), (c), (d), the sample size  $N_T$ , the number of cells when the procedure stops, and the solution time are best fitted by quadratic functions of the dimension of the random vector. For the same initial sample size  $N_0 = 1000$ , the confidence interval on the optimality gap output by SSAM becomes more conservative as the dimension of the random vector,  $d_\xi$ , increases. This is illustrated by the estimates of the relative optimality gaps reported in the last column of Table 5.4. However, one might expect to use larger initial sample sizes as the number of scenarios increases, yielding better estimators.

## 5.6 Summary and Concluding Remarks

In this chapter, we presented numerical results on the sequential procedure SSAM developed in Chapter 4, where at each iteration, a candidate solution is generated by solving a lower-bounding problem derived from Jensen's inequality applied on a partition of the random vector's support. This lower bound, coupled with a stratified sampling-based estimate of the objective function value at the candidate solution that includes a control variate estimator, provides a point estimate of the optimality gap. The procedure stops when this estimate falls below a pre-specified fraction of the sample standard deviation. Otherwise, the partition is refined, the sample size is increased and the procedure repeats. To study the empirical performance of SSAM, we first applied the procedure to an asset allocation problem. Our results indicate that the control variate estimator on a stratified sample significantly reduces variance. Variances with the control variate estimator are reduced by factors of up to 1575 when compared to crude Monte Carlo estimators. The effectiveness of variance reduction increases as the algorithm proceeds or as finer partitions of  $\Xi$  are used. SSAM performs well on the asset allocation problem instance

(a) Sample Sizes ( $N_T$ )(b) Stopping Iteration ( $T$ )

(c) # Cells



(d) Time(s)

Figure 5.1: Dimensionality Study of SSAM

considered, resulting in high empirical coverage probability (100%), few iterations and a relatively small number of cells for the partition of the random vector. In addition, we studied the performance of SSAM as the dimension  $d_\xi$  of the random varies by experimenting on several derived instances of 20TERM with different values of the dimension of the random vector  $d_\xi$ . Our results show that the iteration number when SSAM terminates seem to grow linearly with the dimension of the random vector  $d_\xi$ , whereas the sample size  $N_T$ , the number of cells in the partition at stoppage, and the solution time seem to grow quadratically with  $d_\xi$ .

## CHAPTER 6

## CONCLUSIONS

## 6.1 Summary of Contributions

In this dissertation, we developed Monte Carlo sampling-based algorithms to find solutions of desired quality with a desired probability for various classes of two-stage stochastic programs. These algorithms make use of sampling and deterministic bounding techniques within a sequential framework. We developed two main methods within this framework: the *Fixed-Width Sequential Sampling Methods* and the *Sampling-based Sequential Approximation Methods*.

Fixed-width sequential sampling methods use two independent streams of independent samples with increasing sample sizes are used. One stream is used to generate a candidate solution by solving a sampling approximation problem while the other stream is used to assess the quality of the candidate solution by forming a confidence interval on its optimality gap. The algorithms stop when the confidence interval on the optimality gap plus an inflation factor falls below a pre-specified tolerance  $\varepsilon > 0$ . We developed two procedures under these methods. The fully sequential procedures (FSP) use a sequence of sample sizes from deterministic schedules whereas the sequential procedures with stochastic schedules (SSP) use the current estimates to determine the sample sizes at the next iteration. Empirical results of the application of these procedures on tests problems from the literature show that they perform well in practice, resulting in high coverage probability using moderate sample sizes. We also provided guidelines on how to apply these procedures in practice by experimenting on a large scale two-stage stochastic programming problem.

The sampling-based sequential approximation methods (SSAM), inspired from classical sequential approximation methods, combine both sampling approximation and deterministic bounding approaches for stochastic programs to generate high quality candidate

solutions. For the class of problems considered, Jensen’s inequality is used to solve a lower bounding problem that also generates a candidate solution. An upper bound is obtained through sampling and the procedure lends itself to a proven stopping criterion from classical sequential sampling methods. The procedure stops when the point estimate on the optimality gap falls below a factor relative to its sample standard deviation. These procedures proceed by successively refining a partition of the random vector. We provided rules for partitioning the space of the random vector and studied the magnitude of several variance reduction schemes over crude Monte Carlo in calculating the sample estimates for the optimality gap. SSAM performs well (high coverage probability for relatively small number of iterations and moderate solution time) on an asset allocation test problem. Dimensionality studies on instances of the native 20TERM model shows that key performance measures (e.g. solution time, number of cells in the partition, iteration number at stoppage) of SSAM increase moderately with the dimensionality  $d_\xi$  of the space of the random vector.

The contributions of the work presented in this dissertation can be summarized as follows:

- We provided an algorithmic framework for sampling and bounding approximation approaches for solving stochastic programs. This framework allows the development of two distinctive families of sequential algorithms.
- We developed fixed-width sequential sampling methods that allow the use of confidence interval estimators of the optimality gap as a viable stopping rule in a sequential setting. Rules for increasing the sample size, either through deterministic or stochastic schedules, are established along with key theoretical properties that guarantee convergence, finite stopping and validity results. We provided, as consequences of our theoretical results, a practical approach for selecting a tolerance value  $\varepsilon > 0$  given a sample size budget. Computational results show that these algorithms have good performance in practice.
- We developed sampling-based sequential approximation methods that combine the computationally easy Jensen’s lower bound from classical sequential approxima-

tion with a sampling upper bound from sequential sampling. Computationally effective guidelines are provided for partitioning rules, sample size increases as well schemes for variance reduction that result in tighter confidence intervals on the optimality gap. Key convergence results are established and confirmed through computational experiments on test problems from the literature.

- We created new problem instances from the 20TERM model with varying dimensionality of the random vector. These new problem instances allow appropriate dimensionality study benchmarking for algorithms, such as the ones developed in this dissertation, that attempt to solve large scale two-stage stochastic programs.

## 6.2 Future Research

Several future research directions stemming from the work presented in this dissertation are as follows:

- *Relative optimality gap driven algorithms*: The fixed-width sequential sampling algorithms take an  $\varepsilon$ -value as an input to the procedures. In order to make a valid statement on the quality of the solutions output by the procedures, this supposes that some definite statement can be made about that  $\varepsilon$ -value and  $z^*$ . While some preliminary study could be done in order to determine appropriate values of  $\varepsilon$  that correspond to desired percentage of optimality, the algorithms could be modified to incorporate that determination as an intrinsic part of the procedures described in this dissertation.
- *Bias and variance reduction of estimators*: Both classes of algorithms developed in this dissertation make use of sampling estimates. In the case of SSAM, we make use of stratified sampling and control variates in order to reduce the variance of the optimality gap estimate. We can extend a similar approach for the fixed-width algorithms in forming the confidence interval on the optimality gap. Further, bias reduction techniques can also be considered for both classes of algorithms.

- *Adaptive sequential sampling*: For the fixed-width sequential sampling algorithms, adaptive sampling methods can be developed such that the sample size and type could be determined based on the current progress of the algorithms. The procedure with stochastic sample size schedules is an attempt at that adaptivity. However, it does not explicitly consider the bias or variance, and at what rate bias and variance decrease. Further adaptive methods could explicitly consider these.
- *Empirical comparisons of algorithms*: Certain classes of stochastic models, such as two-stage stochastic linear programs or a class of two-stage stochastic integer programs with uncertainty in the right-hand side, can be solved using either of the class of algorithms presented here. Future investigations can include empirical comparisons of the performance of these algorithms.
- *Use of  $\epsilon$ -optimal solutions*: By construction, the algorithms developed here output  $\epsilon$ -optimal solutions. For the fixed-width sequential algorithms, at least two sets of optimization problems are solved to optimality at each iteration. One approach to consider is to solve these problems not exactly but to within  $\epsilon$  of the (approximated) optimal objective. This is in line with retrospective optimization (Pasupathy, 2010). Allocation of sample sizes within this approach merits further research.

## REFERENCES

- Ahmed, S. and A. Shapiro (2002). The sample average approximation method for stochastic programs with integer recourse. *Optimization Online*. [Http://www.optimization-online.org/](http://www.optimization-online.org/).
- Attouch, H. and R.-B. Wets (1981). Approximation and convergence in nonlinear optimization. In Mangasarian, O., R. Meyer, and S. Robinson (eds.) *Nonlinear Programming 4*, pp. 367–394. Academic Press, New York.
- Baker, S., D. Morton, R. Rosenthal, and L. Williams (2002). Optimizing military airlift. *Operations Research*, **50**(4).
- Bayraksan, G. and D. Morton (2006). Assessing solution quality in stochastic programs. *Mathematical Programming*, **108**, pp. 495–514.
- Bayraksan, G. and D. P. Morton (2011). A sequential sampling procedure for stochastic programming. *Operation Research*, **59**(4).
- Beale, E. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, **17B**, pp. 173–184.
- Birge, J. and J. Dulá (1991). Bounding separable recourse functions with limited distribution information. *Annals of Operations Research*, **30**, pp. 277–298.
- Birge, J. and M. Teboulle (1989). Upper bounds on the expected value of a convex function using gradient and conjugate function information. *Mathematics of Operations Research*, **14**.
- Birge, J. and S. Wallace (1986). Refining bounds for stochastic linear programs with linearly transformed independent random variables. *Operations Research Letters*, **5**, pp. 73–77.
- Birge, J. and R. Wets (1986). Designing approximation schemes for stochastic optimization problems, in particular, for stochastic programs with recourse. *Mathematical Programming Study*, **27**, pp. 54–102.
- Birge, J. and R. Wets (1987). Computing bounds for stochastic programming problems by means of a generalized moment problem. *Mathematics of Operations Research*, **12**, pp. 149–162.
- Birge, J. and R. Wets (1989). Sublinear upper bounds for stochastic programs with recourse. *Mathematical Programming*, **43**, pp. 131–149.

- Bratley, P., B. Fox, and L. Schrage (1987). *A Guide to Simulation*. Springer-Verlag.
- Cariño, D., T. Kent, D. Meyers, C. Stacy, M. Sylvanus, A. Turner, K. Watanabe, and W. Ziemba (1994). The Russell-Yasuda Kasai model: an asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces*, **24**, pp. 29–49.
- Chow, Y. S. and H. Robbins (1965). On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *Annals of Mathematical Statistics*, **36**(2), pp. 457–462.
- Dai, L., C. H. Chen, and J. R. Birge (2000). Convergence properties of two-stage stochastic programming. *Journal of Optimization Theory and Applications*, **106**, pp. 489–509. ISSN 0022-3239. doi:10.1023/A:1004649211111.
- Dantzig, G. (1955). Linear programming under uncertainty. *Management Science*, **1**, pp. 197–206.
- Dantzig, G. and P. Glynn (1990). Parallel processors for planning under uncertainty. *Annals of Operations Research*, **22**, pp. 1–21.
- Dantzig, G. and G. Infanger (1995). A probabilistic lower bound for two-stage stochastic programs. Technical Report SOL 95-6, Department of Operations Research, Stanford University.
- Dokov, S. P. and D. P. Morton (2005). Second-order lower bounds on the expectation of a convex function. *Mathematics of Operations Research*, **30**(3), pp. 662–677. ISSN 0364765X.
- Dupačová, J. (1976). Minimax stochastic programs with nonconvex nonseparable penalty functions. In Prékopa, A. (ed.) *Progress in Operations Research*, pp. 303–316. Mathematica Societatis János Bolyai, Eger, Hungary.
- Dupačová, J. (1987). The minimax approach to stochastic programming and an illustrative application. *Stochastics*, **20**, pp. 73–88.
- Dupačová, J. and R. Wets (1988). Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. *The Annals of Statistics*, **16**, pp. 1517–1549.
- Dupačová, J. (1966). On minimax solutions of stochastic linear programming problems. *Časopis pro Pěstování Matematiky*, **91**, pp. 423–429.
- Edirisinghe, N. (1996). New second-order bounds on the expectation of saddle functions with applications to stochastic linear programming. *Operations Research*, **44**, pp. 909–922.

- Edirisinghe, N. and W. Ziemba (1992). Tight bounds for stochastic convex programs. *Operations Research*, **40**, pp. 660–677.
- Edirisinghe, N. and W. Ziemba (1994). Bounding the expectation of a saddle function with application to stochastic programming. *Mathematics of Operations Research*, **19**, pp. 314–340.
- Eppen, G., R. Martin, and L. Schrage (1989). A scenario approach to capacity planning. *Operations Research*, **37**, pp. 517–527.
- Ermoliev, Y. (1988). Stochastic quasigradient methods. In Ermoliev, Y. and R.-B. Wets (eds.) *Numerical Techniques for Stochastic Optimization*, pp. 141–185. Springer-Verlag, Berlin.
- Fiacco, A. V. and J. Kyparisis (1986). Convexity and concavity properties of the optimal value function in parametric nonlinear programming. *Journal of Optimization Theory and Applications*, **48**(1), pp. 95–126. ISSN 0022-3239.
- Frantzeskakis, L. and W. Powell (1990). A successive linear approximation procedure for stochastic, dynamic vehicle allocation problems. *Transportation Science*, **24**, pp. 40–57.
- Frauendorfer, K. (1992). *Stochastic two-stage programming*. Lecture Notes in Economics and Mathematical Systems 392. Springer-Verlag, Berlin.
- Frauendorfer, K. and P. Kall (1988). A solution method for SLP recourse problems with arbitrary multivariate distributions—the independent case. *Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform.*, **17**(4), pp. 177–205.
- Futschik, A. and G. Pflug (1997). Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms. *European Journal of Operational Research*, **101**, pp. 245–260.
- Gaivoronski, A. (1988). Stochastic quasigradient methods and their implementation. In *Numerical techniques for stochastic optimization*, volume 10 of *Springer Ser. Comput. Math.*, pp. 313–351. Springer, Berlin.
- Gassmann, H. and W. Ziemba (1986). A tight upper bound for the expectation of a convex function of a multivariate random variable. *Mathematical Programming Study*, **27**, pp. 39–53.
- Ghosh, B. and P. Sen (eds.) (1991). *Handbook of Sequential Analysis*. Marcel Dekker, Inc., New York.
- Ghosh, M., N. Mukhopadhyay, and P. K. Sen (1997). *Sequential Estimation*. Wiley, New York.

- Glynn, P. W. and W. Whitt (1992). The asymptotic validity of sequential stopping rules for stochastic simulations. *The Annals of Applied Probability*, **2**(1), pp. 180–198.
- Grant, M. and S. Boyd (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., S. Boyd, and H. Kimura (eds.) *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- Grant, M. and S. Boyd (2011). CVX: Matlab Software for Disciplined Convex Programming, version 1.21. <http://cvxr.com/cvx>.
- Gut, A. (1988). *Stopped Random Walks: Limit Theorems and Applications*. Springer-Verlag, New York.
- Hammersley, J. and D. Handscomb (1964). *Monte Carlo Methods*. Chapman and Hall, London.
- Higle, J. and S. Sen (1991a). Statistical verification of optimality conditions for stochastic programs with recourse. *Annals of Operations Research*, **30**, pp. 215–240.
- Higle, J. and S. Sen (1991b). Stochastic decomposition: an algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, **16**, pp. 650–669.
- Higle, J. and S. Sen (1994). Finite master programs in regularized stochastic decomposition. *Mathematical Programming*, **67**, pp. 143–168.
- Higle, J. and S. Sen (1996a). Duality and statistical tests of optimality for two stage stochastic programs. *Mathematical Programming*, **75**, pp. 257–275.
- Higle, J. and S. Sen (1996b). *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*. Kluwer Academic Publishers, Dordrecht.
- Homem-de-Mello, T. (2003). Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, **13**(2), pp. 108–133.
- Homem-de-Mello, T. (2008). On rates of convergence for stochastic optimization problems under non-i.i.d. sampling. *SIAM Journal on Optimization*, **19**, pp. 524–551.
- Hong, L. and B. Nelson (2005). The tradeoff between sampling and switching: New sequential procedures for indifference-zone selection. *IIE Transactions*, **37**, pp. 623–634.
- Huang, C., W. Ziemba, and A. Ben-Tal (1977). Bounds on the expectation of a convex function of a random variable: with applications to stochastic programming. *Operations Research*, **25**, pp. 315–325.

- Infanger, G. (1992). Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs. *Annals of Operations Research*, **39**, pp. 69–95.
- Jensen, J. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, **30**, pp. 175–193.
- Kall, P. (1986). Approximation to optimization problems: an elementary review. *Mathematics of Operations Research*, **11**, pp. 9–18.
- Kall, P. (1991). An upper bound for slp using first and total second moments. *Annals of Operations Research*, **30**, pp. 267–276.
- Kall, P., A. Ruszczyński, and K. Frauendorfer (1988). Approximation techniques in stochastic programming. In Ermoliev, Y. and R.-B. Wets (eds.) *Numerical Techniques for Stochastic Optimization*, pp. 33–64. Springer-Verlag, Berlin.
- Kall, P. and S. Wallace (1994). *Stochastic Programming*. John Wiley & Sons, Chichester.
- Kaniovski, Y., A. King, and R. Wets (1995). Probabilistic bounds (via large deviations) for the solutions of stochastic programming problems. *Annals of Operations Research*, **56**, pp. 189–208. ISSN 0254-5330. 10.1007/BF02031707.
- Kim, S.-H. and B. Nelson (2006). On the asymptotic validity of fully sequential selection procedures for steady-state simulation. *Operations Research*, **54**(3), pp. 475–488.
- King, A. and R. Rockafellar (1993). Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, **18**, pp. 148–162.
- Kleywegt, A., A. Shapiro, and T. Homem-de-Mello (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, **12**(3), pp. 479–502.
- Kushner, H. and G. Yin (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York.
- Lavenberg, S. S. and P. D. Welch (1981). A perspective on the use of control variables to increase the efficiency of Monte Carlo simulations. *MANAGEMENT SCIENCE*, **27**, pp. 322–335.
- Law, A. and W. Kelton (1982). Confidence intervals for steady-state simulations II: A survey of sequential procedures. *Management Science*, **28**, pp. 550–562.
- Law, A., W. Kelton, and L. Koenig (1981). Relative width sequential confidence intervals for the mean. *Communications in Statistics*, **B10**, pp. 29–39.

- Madansky, A. (1959). Bounds on the expectation of a convex function of a multivariate random variable. *Annals of Mathematical Statistics*, **30**, pp. 743–746.
- Madansky, A. (1960). Inequalities for stochastic linear programming problems. *Management Science*, **6**, pp. 197–204.
- Mak, W., D. Morton, and R. Wood (1999). Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, **24**, pp. 47–56.
- Morton, D. (1998). Stopping rules for a class of sampling-based stochastic programming algorithms. *Operations Research*, **46**, pp. 710–718.
- Morton, D., R. Rosenthal, and L. Weng (1996). Optimization for airlift mobility. *Military Operations Research*, **1**, pp. 49–68.
- Morton, D. and R. Wood (1999). Restricted-recourse bounds for stochastic linear programming. *Operations Research*, **47**, pp. 943–956.
- Nadas, A. (1969). An extension of a theorem of Chow and Robbins on sequential confidence intervals for the mean. *Annals of Mathematical Statistics*, **40**(2), pp. 667–671.
- Nelson, B. L. (1990). Control variate remedies. *Operations Research*, **38**(6), pp. 974–992. ISSN 0030-364X. doi:<http://dx.doi.org/10.1287/opre.38.6.974>.
- Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, **19**, pp. 1574–1609. ISSN 1052-6234. doi:10.1137/070704277.
- Norkin, V., G. Pflug, and A. Ruszczyński (1998). A branch and bound method for stochastic global optimization. *Mathematical Programming*, **83**, pp. 425–450.
- Partani, A., D. P. Morton, and I. Popova (2006). Jackknife estimators for reducing bias in asset allocation. In *Proceedings of the Winter Simulation Conference*.
- Pasupathy, R. (2010). On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research*, **58**(4-Part-1).
- Pflug, G. C. (1988). Stepsize rules, stopping times and their implementations in stochastic quasigradient algorithms. In Ermoliev, Y. and R. J.-B. Wets (eds.) *Numerical Techniques for Stochastic Optimization*. Springer Verlag, Berlin.
- Polak, E. and J. Royset (2008). Efficient sample sizes in stochastic nonlinear programming. *Journal of Computational and Applied Mathematics*, **217**, pp. 301–310.

- Richter, W. (1965). Limit theorems for sequences of random variables with sequences of random indices. *Theory of Probability and its Applications*, **10**, pp. 74–84.
- Robbins, H. and S. Monro (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, **22**, pp. 400–407.
- Ruszczynski, A. (1986). A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, **35**, pp. 309–333.
- Ruszczynski, A. and A. Świetanowski (1997). Accelerating the regularized decomposition method for two stage stochastic linear problems. *European Journal of Operational Research*, **101**, pp. 328–342.
- Sen, S., R. Doverspike, and S. Cosares (1994). Network planning with random demand. *Telecommunication Systems*, **3**, pp. 11–30.
- Shapiro, A. (1989). Asymptotic properties of statistical estimators in stochastic programming. *The Annals of Statistics*, **17**, pp. 841–858.
- Shapiro, A. (1991). Asymptotic analysis of stochastic programs. *Annals of Operations Research*, **30**, pp. 169–186.
- Shapiro, A. (2003). Monte Carlo sampling methods. In Ruszczyński, A. and A. Shapiro (eds.) *Handbooks in Operations Research and Management Science, Volume 10: Stochastic Programming*, pp. 353–425. Elsevier.
- Shapiro, A. and T. Homem-de-Mello (2000). On rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, **11**, pp. 70–86.
- Shapiro, A. and A. Nemirovski (2005). On complexity of stochastic programming problems. In Jeyakumar, V. and A. Rubinov (eds.) *Continuous Optimization: Current Trends and Applications*, pp. 111–144. Springer, Berlin.
- Topaloglu, H. (2009). A tighter variant of Jensen’s lower bound for stochastic programs and separable approximations to recourse functions. *European Journal of Operational Research*, **199**(2), pp. 315–322.
- Verweij, B., S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro (2003). The sample average approximation method applied to stochastic vehicle routing problems: a computational study. *Computational and Applied Optimization*, **24**, pp. 289–333.