2016

Monte Carlo Approx. Methods for Stochastic Optimization

John Fowler

Senior Thesis in Mathematics

Monte Carlo Approx. Methods for Stochastic Optimization

Author:
John Fowler

Advisor:
Dr. Mark Huber

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

March 9, 2016

Abstract

This thesis provides an overview of stochastic optimization (SP) problems and looks at how the Sample Average Approximation (SAA) method is used to solve them. We review several applications of this problem-solving technique that have been published in papers over the last few years. The number and variety of the examples should give an indication of the usefulness of this technique. The examples also provide opportunities to discuss important aspects of SPs and the SAA method including model assumptions, optimality gaps, the use of deterministic methods for finite sample sizes, and the accelerated Benders decomposition algorithm. We also give a brief overview of the Sample Approximation (SA) method, and compare it to the SAA method.

Contents

Chapter 1

Introduction

Let us begin by considering a generic *optimization problem* in standard form:

$$\begin{aligned} \underset{x \in S}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_i(x) = 0,\ i = 1, \ldots, p \end{aligned}$$

We say that $f(x)$ is the *objective function*, and the $g_i(x) \leq 0$ and $h_i(x) = 0$ inequalities and equations are a set of *constraints*. Given this set of constraints, there will be a set of feasible solutions, S, to the program. The goal of the optimization problem is to find the element of S that minimizes the objective function.

We now consider a *stochastic optimization problem (SP)*, the area of optimization that will be focused on for the remainder of this thesis. An SP is an optimization problem that takes randomness directly into account. The uncertainty in the problem can show up via random variables in the objective function, the constraint(s), or some combination of the objective function and constraints. Typically, this randomness is accounted for using probability distributions to estimate the way that the variables behave. We now consider an SP in generic form. Note that the program allows for both linear and nonlinear constraints and objective function:

Problem 1 *Generic Stochastic Optimization Problem*

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

Let us first define the elements of (1.1):

- P is a probability distribution
- ξ is a random vector with probability distribution P
- S is a finite set of feasible solutions, and x is an element of S
- $G(x, \xi)$ is a real-valued function of x and ξ
- $g(x) = \int G(x, \xi) P(d\xi)$ is the expected value (or mean) of that function

We are especially interested in SPs that meet the following criteria:

1. The set of feasible solutions, S, is very large.
2. The expected value function, $g(x)$:
   - cannot be calculated easily with numerical methods.
   - cannot be written in closed form.
3. Given x and ξ, the function $G(x, \xi)$ is easy to compute.

If the first two of these criteria are met, then solving this type of optimization problem becomes very difficult to do with numerical methods. In addition, if the final criterion is met, then solving $G(x, \xi)$ via simulation is relatively easy. This ability to use simulation allows us to utilize methods where we can approximate the expected value function to come up with good estimates for the SP.

1.1 Two-Stage Stochastic Optimization

One of the more common types of stochastic optimization problems is the *Two-Stage Stochastic Optimization Problem (TSSP)*. In such a program, we break the $G(x, \xi)$ function into two separate functions:

- $f(x)$ is the *first stage* and is only a function of x so it has no randomness
- $h(x, \xi)$ is the *second stage* and is a solution to a secondary optimization problem that can be solved after obtaining a realization of the uncertain data, ξ, from the probability distribution P

Let us define several elements of a generic TSSP before continuing:

- x is the First Stage decision variable
- y is the Second Stage decision variable
- ξ is a random vector that contains the random elements T, W, h, and q.

We now outline a generic form TSSP. Note that the program allows both linear and nonlinear objective functions in the first stage and second stage problems, as well as linear and nonlinear constraints in the first stage. However, the constraints of the second stage problem are often linear to facilitate the ease of solving these types of problems:

Problem 2 *Generic Two-Stage Stochastic Optimization Problem*

$$\min_{x \in S} \{g(x) := f(x) + \mathbb{E}_P[h(x, \xi)]\} \tag{1.2}$$

where $h(x, \xi)$ is the optimal value of the following second stage problem:

$$\begin{aligned} \min_{y} \quad & q(y, \xi) \\ \text{subject to} \quad & Tx + Wy = h \\ & y \geq 0 \end{aligned} \tag{1.3}$$

By breaking the SP into two distinct stages, we are able to make decisions based only on information currently available at the time of the decision making. During the First Stage, we optimize the function of the first stage decision variable and the expected value of the second stage problem. We then obtain a realization of ξ and proceed to optimize the $h(x, \xi)$ function during the Second Stage.

1.2 Literature and Examples

There exists much literature on the topic of stochastic optimization problems and how to solve them. Two such examples are a 2002 paper by Kleywegt et al. [8] and, more recently, a 2014 paper by Homem-de-Mello and Bayraksan [3]. For background on stochastic programming one can reference the book

by Shapiro et al. [12]. An article with a focus on two-stage stochastic programming can be found in the paper by Shapiro and Homem-de-Mello [11].

As we briefly discussed after defining a generic SP, these problems are often solved using simulation techniques. One of the interesting aspects of using simulation to solve these types of stochastic programs is the wide variety of applications. One such application is in determining reasonable transportation routes and optimal resource allocations in the case of national or regional emergencies, as Chunlin and Liu [2] did for China in response to events like the SARS outbreak of 2003 and the Wenchuan earthquake of 2008. Another common area of application is the commercial realm. One example of this is included in a paper by Xu and Zhang [13] in which they determine an optimal allocation of electricity in a market where options are used as a contract between buyer and seller. The dissimilarity of these two examples is representative of the variability in applications for the use of the SAA method in solving SPs.

The remainder of the thesis is organized as follows:

- In Section 2, we detail the most common simulation method used in solving stochastic optimization problems, the sample average approximation (SAA) method. This section relies on Fu's [5] review of simulation methods to solve optimization problems, Shapiro's [10] work on the use of Monte Carlo methods, including SAA, to solve stochastic optimization problems, and the guiding overview by Kim et al. [7] of how and when to use SAA.

The next four sections are examples of applications of SAA in solving stochastic optimization problems:

- An example from El-Rifai et al. [4] in shift scheduling within emergency departments and some typical assumptions in these types of models can be found in Section 3.

- Section 4 discusses an example from Ayvaz et al. [1] of electronics recycling and the use of optimality gaps.

- The next section overviews a vehicle routing application from Kenyon and Morton [6] and touches on how deterministic methods can be used for small cardinality of n.

- In Section 6, we discuss supply chain applications found in Santoso et al. [9] and the use of the accelerated Benders decomposition algorithm to help solve problems with many scenarios.

From here we back out of the examples:

- Section 7 gives a brief comparison of the stochastic approximation method with SAA, as elucidated in Kim et al. [7] and Shapiro et al. [12].
- Finally, we end with some concluding remarks in Section 8.

Chapter 2

The Sample Average Approximation Method

As we touched on at several points during the introduction, simulation is one of the most common ways that people use to solve the types of stochastic optimization problems of the forms shown in Problems 1 and 2. The simulation-based solutions that we will discuss here are under the category of computational algorithms known as *Monte Carlo algorithms*. In general, we can define a Monte Carlo algorithm as any algorithm with a deterministic running time, but a random output. We now outline a generic version of basic Monte Carlo simulation.

Let X be a random variable with mean μ. To estimate μ, follow these steps:

1. Draw X_1, X_2, ..., $X_n \underset{\leftarrow}{iid} X$
2. Output $\bar{x} = (X_1 + X_2 + ... + X_n)/n$

The output $\bar{x}$ is an estimate for μ. Also, the error in this estimate is approximated by $SD(X)/\sqrt{n}$, where $SD(X)$ is the standard deviation of the random variable X. This nature of this error relationship is significant because it means that the error decreases as n increases. This allows us to often produce relatively accurate estimates via Monte Carlo algorithms simply by simulating a large number of trials.

While Monte Carlo simulation can be used to help solve a variety of problems, including numerical integration and modeling probability distributions

to name a few, here we are focusing on the algorithm's application within the field of numerical optimization (especially for those SPs meeting the three criteria we outlined in the introduction). In particular, we are focusing on the use of a Monte Carlo method known as *Sample Average Approximation (SAA)*.

The basic idea of SAA is to take an independent, identically distributed random sample from the probability distribution P of our generic stochastic optimization problem of the form (1.1) to come up with a realization, ξ_i, of the random vector ξ. We repeat this process some relatively large number N times so that we have N different realizations: ξ_1, ..., ξ_N. We then estimate $g(x) := \mathbb{E}_P[G(x, \xi)]$ with:

$$\hat{g}_N(x) := \frac{1}{N} \sum_{i=1}^{N} G(x, \xi_i) \tag{2.1}$$

In doing this, we have effectively removed the randomness from the problem by taking our N realizations as given. This means that the optimal solution to our stochastic optimization problem can be approximated by the optimal solution to the deterministic optimization problem:

Problem 3 *Generic Sample Average Approximation Problem*

$$\min_{x \in S} \{\hat{g}_N(x) := \frac{1}{N} \sum_{i=1}^{N} G(x, \xi_i)\} \tag{2.2}$$

where $\xi_1, ..., \xi_N$ is a random sample from probability distribution P.

One piece of information that is important to note about SAA is that it is not an algorithm in and of itself. The sole purpose of SAA is to replace the original SP with its sampling approximation of the form (2.1). However, the generic algorithm for solving these types of problems, while not very specific, is easy to write down. We outline it here for completeness:

Problem 4 *Algorithm for Solving Generic SPs Using SAA*

1. *Start with a stochastic optimization problem of the form (1.1).*
2. *Draw a random set of N realizations $\xi_1, ..., \xi_N$ from P.*

3. *Use this set of realizations to reformulate the SP into a problem of the form (2.2).*

4. *Use any of a number of deterministic methods to solve this reformulated problem to come up with specific estimates for both the optimal element of the feasible set and optimal value of the original SP.*

In the following four chapters, we will see various applications of the SAA method for solving stochastic optimization problems. Each of these examples will provide insight into some of the many other interesting aspects of the SAA approach that were not touched in this section, including typical assumptions within the model, the use of optimality gaps to ensure we find an appropriately optimal solution, and how the number of scenarios in a problem might change the way we go about solving that problem.

Chapter 3

Model Assumptions

One of the areas of application for solving SPs by using the SAA approach is in shift scheduling, especially in an important and complicated setting like an emergency department. Increased demand and scarcity of resources in the health industry has led to issues with overcrowding and unacceptable waiting times in many French emergency departments during recent years. A paper written in 2014 by El-Rifai et al. [4] worked to address this problem.

We will outline a simplified version of the SP from this paper, but first we must define a few variables:

- Let x be an employee schedule.
- Let $1 \leq t \leq T$ be a time period during the day.
- Let $1 \leq q \leq Q$ be the queue numbers of the Q nurses working at any particular time t.
- Let ξ be a random vector drawn from probability distribution P that models patient entry to the emergency room.
- Let $W_{t,q}(x, \xi)$ be a real-valued function that gives the number of patients waiting in queue q at time t given some employee schedule x and random patient entry vector ξ.
- Let $g_i(x) \leq 0$ for $i = 1, \ldots, m$ be a set of m constraints that ensure employee schedule x allows for adequate resource allotment throughout the emergency room.

We now have the information needed to consider this example SP:

Problem 5 *Emergency Room Scheduling Stochastic Program Example*

$$\begin{aligned} \underset{x \in S}{\textit{minimize}} \quad & \mathbb{E}_P[\sum_{t=1}^{T} \sum_{q=1}^{Q} W_{t,q}(x, \xi)] \\ \textit{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \end{aligned} \tag{3.1}$$

3.1 Building the Model

The authors chose to model the emergency department environment using a mixed-integer SP and went about solving the SP using the SAA method. While the use of stochastic programming and the sample average approximation approach may provide a generic structure to the problem, the specific assumptions made are of the utmost importance to the quality and fit of the model to the reality of the ongoings at the hospital. As such, it is very important to choose appropriate assumptions and to choose metrics that accurately reflect the problem at hand.

In this particular example, the authors decided to address the problem of overcrowding in the emergency department by finding the shift distribution for employees that minimizes patient waiting time. This choice is important, as they had to choose variables that they had control over (i.e. employee shift distribution) and an objective function that was functional as a metric of how to solve the issue of overcrowding (i.e. minimize patient waiting time). Without these targeted choices accurately reflecting the environment of the emergency department, the math in the model would be unlikely to capture and help solve the problem of overcrowding.

The point that we would like to get across is that the assumptions and variable choices of a model are important. Without carefully considering and testing these attributes, it can be very easy to pick ones that do not accurately reflect the important aspects of the environment that a person wishes to model.

Many of the decisions regarding model assumptions that must be made have to do with the level of detail and specificity that are put into those assump-

tions. The authors must be careful to provide enough detail so that they can strike the balance between a system that accurately models the real-world environment but also solves the problem efficiently. This can be seen in the hospital example as the authors chose to divide patients into two classes, those who require auxiliary exams and those who do not. The authors felt that this was an appropriate level of classification, as these two types of patients have distinct trajectories within the emergency department. However, the authors did not feel the need to separate the patients any further, as they reasoned that having more specific classifications would not improve the quality of the solution enough to justify the decrease in efficiency.

This is just one example showing the importance of modeling assumptions. Once these assumptions are properly tested and validated through means like numerical experiments, we can begin to understand the potential accuracy of our model. Once we have an appropriate model we can begin to look into some of the more interesting aspects of the SAA method, as we will do in the next few chapters.

Chapter 4

Optimality of Solutions

Another common area of application for using the SAA method to solve SPs is in the field of reverse logistics within a supply chain. There is a high amount of uncertainty involved in the recycling of electronics in reverse logistics; there are often multiple unknown factors in a robust electronic recycling model, including the quantity of items that will be returned, the conditions of those items, and the transportation costs associated with returning those items. This uncertainty often makes setting up these types of problems in the form of an SP a strong approach. One specific example of this can be found in the paper by Ayvaz et al. [1], where the authors present a two-stage SP model for recycling materials, which they choose to solve using a variation of the SAA method. The authors then test their model on a case study of a Turkish electronic recycling firm.

Before presenting a simplified version of the two-stage SP created by Ayvaz et al., we must first define several variables:

- Let x denote some combination of collecting, sorting, and recycling centers that the firm is considering opening. Note that the number and locations of the centers varies with each x.
- Let ξ be a draw from a probability distribution P that models the uncertain income and transportation and processing costs associated with x.
- Let $h_i(y) = 0$ for $i = 1, \ldots, m$ be a set of m constraints associated with properly regulating the flow of product and the capacity of the centers.

- First Stage Costs (certain)
  $F(x)$ = fixed costs of building centers
- Expected Second Stage Costs (uncertain)
  $I(x, \xi)$ = income
  $T(x, \xi)$ = transportation costs
  $P(x, \xi)$ = processing costs of collecting, sorting, and recycling

We can now consider this type of profit-maximizing two-stage stochastic program:

Problem 6 *Profit-Maximizing Two-Stage Stochastic Optimization Problem*

$$\max_x \{\mathbb{E}_P[q(x, \xi)] - F(x)\} \tag{4.1}$$

where $q(x, \xi)$ *is the optimal value of the following second stage problem:*

$$\begin{aligned} \max_y \quad & I(y, \xi) - T(y, \xi) - P(y, \xi) \\ \textit{subject to} \quad & h_i(y) = 0,\ i = 1, \ldots, m \\ & y \geq 0 \end{aligned} \tag{4.2}$$

While this problem was specifically tailored to the electronic equipment recycling example, one can easily see how it could serve as an example of a more generic profit-maximizing problem. In more general terms, the goal of the first stage problem is to maximize an estimated income minus some estimated variable costs and any known fixed costs over the set of feasible solutions. The estimation of the income and variable costs is determined by finding the optimal solution to the second stage problem.

4.1 Optimality Gaps

One of the most interesting aspects of this paper is the use of an optimality gap, a tool used in conjunction with the SAA method, to measure the quality of a solution to a stochastic program. Optimality gaps allow the quality of a solution to be held to a certain standard. This is done by solving a problem of the form (2.2) until a specified stopping criterion is met. This stopping criterion is often of a form where the sample size N must be increased by a

certain amount until the estimated optimality gap is less than some appropriate value.

We will now outline one of the more commonly used procedures for estimating the optimality gap and show how the gap gives insight into the quality of the candidate solution $\hat{x} \in S$ given by the SAA approach for a sample of size N. In order to do this, we must define several important terms and state a lemma related to finding the upper bound of the optimality gap:

- Let v^* be the optimal value of a problem of the form $\min_{x \in S}\{g(x) := \mathbb{E}_P[G(x, \xi)]\}$ from Problem (1.1).
- Let v_N be the optimal value of the equivalent SAA problem of the form $\min_{x \in S}\{\hat{g}_N(x) := \frac{1}{N}\sum_{i=1}^{N} G(x, \xi_i)\}$ from Problem (2.2).
- The *optimality gap* of a candidate solution $\hat{x}$ is equal to $g(\hat{x}) - v^*$.

Lemma 4.1 $\mathbb{E}[v_N] \leq v^*$

Proof $\mathbb{E}[v_N] = \mathbb{E}[\min_{x \in S}\{\frac{1}{N}\sum_{i=1}^{N} G(x, \xi_i)\}] \leq$

$\min_{x \in S} \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} G(x, \xi_i)] = \min_{x \in S} \frac{1}{N}\mathbb{E}[\sum_{i=1}^{N} G(x, \xi_i)] =$

$\min_{x \in S} \frac{1}{N} N\mathbb{E}[G(x, \xi)] = \min_{x \in S} \mathbb{E}[G(x, \xi)] =$

$\min_{x \in S} g(x) = v^*$ ∎

We now have the tools to understand the optimality gap for a specific candidate solution $\hat{x} \in S$ produced by the SAA method. First note that the value of v^* is unknown. However, this value can be estimated via a number of sampling and bounding techniques. In this case we choose to estimate the optimality gap using a technique called the *Multiple Replications Procedure* (MRP) outlined in a paper by H. de Mello and Bayraksan [3]:

Lemma 4.1 allows us to estimate an upper bound of $g(\hat{x}) - v^*$ as

$$G_N(\hat{x}) := \hat{g}_N(\hat{x}) - v_N.$$

Then we can use a realization of N draws from ξ, along with the fact that $v_N = \min_{x \in S}\{\hat{g}_N(x)\}$, to estimate $G_N(\hat{x})$ as

$$\hat{G}_N(\hat{x}) := \hat{g}_N(\hat{x}, \xi_1, ..., \xi_N) - \min_{x \in S}\{\hat{g}_N(x, \xi_1, ..., \xi_N)\}.$$

We can then generate k different realizations of ξ and repeat this process for each of those k realizations to come up with k different values of $\hat{G}_N(\hat{x})$. We then find the average of these k values to come up with our final estimator of the optimality gap:

$$\bar{G}(\hat{x}) := \frac{1}{k} \sum_{i=1}^{k} \hat{G}_N^i(\hat{x})$$

If this estimate of the optimality gap is sufficiently small, then the candidate solution can be deemed sufficiently optimal. If the gap is too large, a new candidate solution can be found using the SAA method with a larger value of N, and the optimality gap of the new candidate solution can then be estimated.

Despite the need to solve k different SAA problems to come up with $\bar{G}(\hat{x})$, the MRP is widely used to estimate the optimality gap due to its flexibility. Problems can contain continuous or discrete decision variables, nonlinear terms in the constraints or the objective function, and functions do not need to be convex. This flexibility allows the MRP the ability to be applied to a large variety of problems.

Chapter 5

Small Cardinality of Scenarios

Stochastic vehicle routing is another type of problem that is often modeled using an SP and solved using the SAA method. The typical stochastic vehicle routing problem takes some number of vehicles and optimizes their routes given a goal of servicing some number of locations under random service times and travel times.

The routes are often outlined by a directed graph of N nodes and A arcs, and the stochastic elements (like the travel and services times) are often assumed to have known distributions. Usually these optimization problems deal with objective functions related to completion time, which is defined as the time that elapses from the first vehicle leaving the point of origin to the last vehicle returning to that point after completing their service route.

One interesting example of stochastic vehicle routing is outlined in a journal paper by Kenyon and Morton [6]. In this case, the authors set up and solved two different models of the same vehicle routing problem. The two models were identical other than their respective objective functions, one of which minimizes the completion time for the fleet of vehicles while the other maximizes the probability of completing the task by some desired completion time T.

Which of these objective functions is most appropriate depends on the problem at hand; for example, if the fleet of vehicles is leased and needs to be returned to the origin by a certain time before some substantial late fee is incurred, then maximizing the probability of completing by that time might

make more sense. These details and hypotheticals should serve as a reminder of the importance of customizing these types of generic models to the problem at hand.

5.1 Finite Sample Size

One of the details that this paper addresses is the idea of altering the solution method to the problem depending on the size of sample space. Specifically, the authors indicate that if the cardinality of the sample size is small enough (i.e. finite), then the stochastic program can be rewritten in an equivalent deterministic form.

In order to outline the differences between a generic stochastic vehicle routing problem and its deterministic equivalent we must first define a few useful terms:

- Let $\tilde{c}$ be the vector for random travel times.
- Let $\tilde{\tau}$ be the vector for random service times.
- Let the random vector $\xi = (\tilde{c}, \tilde{\tau})$ be composed of all of the model's random elements.
- Let $(i, j) \in A$ be the set of arcs and $i \in N$ be the set of nodes in the directed graph G used to model the possible routes and locations to be serviced.
- Let x_{ijk} be a binary decision variable equal to 1 if vehicle k includes the arc (i, j) as part of its route, and equal to 0 otherwise.
- Let u_{ik} be a binary decision variable equal to 1 if vehicle k services node i during its route, and equal to 0 otherwise.
- Let W be the set of values that the decision vector (x, u) can take on to satisfy the constraints of the SP.
- Finally, let $h(W, \xi) = \max_{k \in K}(\sum_{(i,j) \in A} \tilde{c}_{ijk} x_{ijk} + \sum_{i \in N} \tilde{\tau}_{ik} u_{ik})$ be a measure of the completion time T.

We can now consider a generic stochastic vehicle routing problem:

Problem 7 *Generic Stochastic Vehicle Routing Problem (GSVRP)*

$$\min_{(x,u)\in W} \quad \mathbb{E}[h(W,\xi) = \max_{k\in K}(\sum_{(i,j)\in A} \tilde{c}_{ijk}x_{ijk} + \sum_{i\in N} \tilde{\tau}_{ik}u_{ik})] \tag{5.1}$$

Finding the solution to this problem essentially entails minimizing the expected value of the completion time while still ensuring that x and u satisfy the specific constraints of the problem.

Now, assuming that ξ is finite with sample size Ω, we can write:

- $\xi = (\xi_1, ..., \xi_\Omega)$ where $\xi_\omega = (\tilde{c}^\omega, \tilde{\tau}^\omega)$ with corresponding probability mass function $p_\omega = P(\xi = \xi_\omega)$.
- We also introduce the continuous decision variable θ_ω, which gives the length of the longest route for sample ω.
- Finally, let V be the set of values that the decision vector (x, u, θ) can take on to satisfy the constraints of the linear program.

Using this information, we can rewrite Problem 7 as a deterministic integer linear program:

Problem 8 *Deterministic Equivalent to GSVRP*

$$\begin{aligned} \min_{(x,u,\theta)\in V} \quad & \sum_{\omega\in\Omega} p_\omega\theta_\omega \\ \textit{subject to} \quad & \theta_\omega \geq (\sum_{(i,j)\in A} \tilde{c}^\omega_{ijk}x_{ijk} + \sum_{i\in N} \tilde{\tau}^\omega_{ik}u_{ik}), \\ & k \in K, \omega \in \Omega \end{aligned} \tag{5.2}$$

The deterministic equivalent problem can then be solved using a branch-and-bound algorithm. However, this approach is often not practical because there may be a very large number of constraints. If that is the case, we can instead solve a sequence of relaxations of the deterministic equivalent problem using what is known as a branch-and-cut approach. Details of this type of algorithm can be found in the paper by Kenyon and Morton [6]. We

will also provide details of another type of solution method, the accelerated Benders decomposition algorithm, in the following chapter.

While only able to handle a relatively modest sample size and potentially computationally unwieldy, the deterministic equivalent approach provides certain advantages by allowing us to find an exact solution to the problem, rather than estimating a solution as we do when solving using the SAA method. It is important to keep these advantages and limitations in mind when deciding how to solve a problem of this variety.

Chapter 6

Large Number of Scenarios

In the previous chapter, we considered the use of a deterministic equivalent problem in solving the generic stochastic vehicle routing problem. In this chapter, we will again consider the use of deterministic equivalents, this time in relation to a generic two-stage stochastic linear programming problem. We will introduce a solution method known as an accelerated Benders decomposition algorithm outlined in a paper on supply chain configuration applications by Santoso et al. [9]. The advantage of this algorithm is that it is able to quickly find high quality solutions to stochastic programs with very large numbers of scenarios.

First consider the generic form of a two-stage stochastic linear program. This is identical to Problem 2 from Chapter 1 with the extra stipulation that the objective functions in each stage must be linear:

Problem 9 *Generic Two-Stage Stochastic Linear Optimization Problem*

$$\min_{x \in S} \{g(x) := c^T x + \mathbb{E}_P[h(x, \xi)]\} \tag{6.1}$$

where $h(x, \xi)$ is the optimal value of the following second stage problem:

$$\begin{array}{ll} \min\limits_{y} & q^T y \\ \textit{subject to} & Tx + Wy = h \\ & y \geq 0 \end{array} \tag{6.2}$$

Now we will formulate this two-stage problem as its deterministic equivalent, just as we did with the vehicle routing problem in the previous chapter. It

is important to recall that each $\xi_i = (T_i, W_i, h_i, q_i)$ is drawn with probability p_i from probability distribution P, and the ξ_i for $1 \leq i \leq N$ encapsulate all of the randomness in the problem.

Problem 10 *Deterministic Equivalent of Stoch. Two-Stage Linear Program*

$$\begin{aligned} \min_{x,y} \quad & c^T x + \sum_{i=1}^{N} p_i q_i^T y \\ \textit{subject to} \quad & T_i x + W_i y = h_i \\ & y \geq 0 \end{aligned} \tag{6.3}$$

6.1 Benders Decomposition Algorithm

Now that we have outlined the deterministic equivalent of the generic stochastic two-stage linear program, we can get into detail about the accelerated Benders decompostion algorithm and how it is used to solve a problem that can be written this way.

The basic premise of this algorithm is to use dual information from a series of sub-problem linear programs to compute optimality cuts. These optimality cuts serve to improve our estimated solution by helping to tighten the gap between our lower and upper bounds until we have a small enough gap to conclude optimality. This allows us to quickly find a solution to the SAA problem when we have a very large number of scenarios.

- Step 1: Set the iteration counter to $j := 0$. Also set $UB := \infty$ as an upper bound and solve to find a lower bound

$$LB := \min_{x \in S}(c^T x) \tag{6.4}$$

  Let $\hat{x}^j$ be the optimal solution to the above problem.

- Step 2: For $i = 1, ..., N$ solve the sub-problems associated with x^j and ξ^i

$$\begin{aligned} h(x^j, \xi^i) := \min_{y} \quad & q_i^T y \\ \text{subject to} \quad & T_i x + W_i y = h_i \\ & y \geq 0 \end{aligned} \tag{6.5}$$

Let $\hat{f}_N(x^j) := c^T x^j + \frac{1}{N}\sum_{i=1}^{N} h(x^j, \xi^i)$. If $\hat{f}_N(x^j) < UB$, then let $UB := \hat{f}_N(x^j)$ and let $\hat{x} := \hat{x}^j$ be the current candidate solution.

- Step 3: If $UB - LB < \delta$ for some pre-defined $\delta \geq 0$, return UB as the optimal objective value and $\hat{x}$ as the optimal solution. If not, continue to Step 4.

- Step 4: For $i = 1, ..., N$, let y^i be the optimal values and π^i be the optimal dual values for each of the sub-problems associated with x^j and ξ^i in Step 2. Now solve the following master problem to find a new lower bound

$$\begin{aligned} LB := \min_{x,y} \quad & c^T x + \theta \\ \text{subject to} \quad & \theta \geq \sum_{i=1}^{N} p_i(-\pi^l[T_i x + W_i y^l - h_i]),\ \forall l = 1, ..., j \end{aligned} \tag{6.6}$$

  Now let $j := j + 1$ and return to Step 2.

By adding more constraints as it moves closer to a solution of some pre-defined level of optimality, this cutting plane algorithm provides a relatively efficient way to find a solution even to some of the larger finite stochastic programs.

Chapter 7

The SA Method

We now briefly consider the *Sample Approximation* (SA) method for solving SPs, one of the most popular alternatives to the SAA method. Rather than solving an SP by estimating the expected value of the objective function via simulation, the SA method takes an iterative and recursive approach. There is an established literature on the subject and we will use works by Kim et al. [7] and Shapiro et al. [12] in our short review of the topic.

Consider an unconstrained stochastic optimization problem in $\mathbb{R}^d$. The SA algorithm is an example of recursion that requires:

- An initial point $\tilde{X}_0$.
- A positive gain sequence $\{a_n\}$ where $n \geq 0$.
- A sequence of vectors $\{Z_n\} \in \mathbb{R}^d$ where $n \geq 0$ and Z_n is an approximation of $\bigtriangledown f(X_n)$. With this type of problem, we often use $Z_n := \bigtriangledown f(\tilde{X}_n, \xi_n)$ as our estimate.

The output of the algorithm is a sequence of points $\{\tilde{X}_n\} \in \mathbb{R}^d$ for $n \geq 0$. The following is the basic recursive structure of the SA approach applied to a generic stochastic minimization problem for some number of iterations n:

Problem 11 *Sample Approximation Method*

$$\tilde{X}_{n+1} = \tilde{X}_n - a_n Z_n \tag{7.1}$$

$\tilde{X}_n$ is our estimate of the optimal solution to the SP after the n^{th} iteration.

The book by Shapiro et al. [12] demonstrates that the sample size guaranteeing an ϵ-optimal solution with a certain probability p for a typical SP is very similar for both the SAA method and the SA method. However, the authors also point out that the cost complexity of solving the problem is much higher for the SAA method, meaning that for the sample sample size N, it will take less computation time for the SA method to achieve a similarly optimal solution to that produced by the SAA method.

It is also important to note that the performance of the recursion in the SA method is heavily dependent on the smoothness of $f(.)$ as well as the choice of the gain sequence $\{a_n\}$. We believe that this, along with the relative simplicity and ease of use of the SAA method, may explain our observation that the SAA method is often preferred in practicical application (at least by the authors we have read) to the SA method.

Chapter 8

Conclusion

We have attempted to provide an overview of some of the most interesting points in regards to solving stochastic optimization problems using the sample average approximation method. We found examples of the application of this approach within fields such as health care, supply chain, and vehicle routing. These examples were used to highlight details of the SPs, like the importance of understanding your model and establishing the proper assumptions. The examples also highlighted details of the SAA method, including the use of optimality gaps to ensure high quality solutions. We also looked at the usefulness of deterministic equivalencies and the Benders decomposition algorithm. Finally, we briefly touched on the stochastic approximation (SA) method, and explored why it seems to be less prevalent in application literature than the SAA approach. We hope that this exposition helped to further general understanding of the flexibility and prevalence of the SAA method as a problem solving technique.

The main sources for my thesis are [3] and [8]. I use these as the backbone and focus of my expository paper, as they tie together most of the ideas that I want to discuss, namely the manner in which the Sample Average Approximation method is generally used to solve stochastic optimization problems. I use [12], [11], [5], [10], and [7] as the mathematical foundation for the concepts discussed in the two main sources. More specifically, I use [11] to focus on two-stage stochastic optimization problems, [12] for background on stochastic programming, and [5], [10], and [7] for background on Monte Carlo Methods and the Sample Average Approximation method. I use [2] and [13] as initial examples of these concepts in the introduction. Finally, I use [1], [6], [4], and [9] as my four main application examples to demonstrate and discuss the wide variety of applications of using the SAA for stochastic optimization problems.

Bibliography

[1] B. Ayvaz, B. Bolat, and N. Aydin. Stochastic reverse logistics network design for waste of electrical and electronic equipment. *Resources, Conservation and Recycling*, 2015.

[2] D. Chunlin and Y. Liu. Sample average approximation method for chance constrained stochastic programming in transportation model of emergency management. *Systems Engineering Procedia*, 5:137–143, 2012.

[3] T. H. de Mello and G. Bayraksan. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.

[4] O. El-Rifai, T. Garaix, V. Augusto, and X. Xie. A stochastic optimization model for shift scheduling in emergency departments. *Health Care Management Science*, 18(3):289–302, 2015.

[5] M. C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.

[6] A. S. Kenyon and D. P. Morton. Stochastic vehicle routing problems with random travel times. *Transportation Science*, 37(1):69–82, 2003.

[7] S. Kim, R. Pasupathy, and S. G. Henderson. A guide to sample average approximation. *Handbook of Simulation Optimization*, 2015.

[8] A. J. Kleywegt, A. Shapiro, and T. H. de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[9] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115, 2005.

[10] A. Shapiro. Monte carlo sampling methods. *Handbooks in Operations Research and Management Science*, 10:353–425, 2003.

[11] A. Shapiro and T. H. de Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81:301–325, 1998.

[12] A. Shapiro, D. Dentcheva, and A. Ruszczyniski. *Stochastic Programming: Modeling and Theory*. SIAM Series on Optimization, 2009.

[13] H. Xu and D. Zhang. Stochastic nash equilibrium problems: sample average approximation and applications. *Computational Optimization and Applications*, 55:597–645, 2013.