SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATION RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ENHANCED ALGORITHMS FOR STOCHASTIC PROGRAMMING

by

Alamuru S. Krishna

TECHNICAL REPORT SOL 93-8

September 1993

Research and reproduction of this report were partially suported by the Boeing Contract W297550; the Department of Energy Grant DE-FG03-92ER~~5116~~; NSF Grant ECS-8906260 and the Electric Power Research Institute Contract RP8010-09. 25116

Any opinions, findings, and conclusions or recommendations expressed in this puplication are those of the author(s) and do NOT necessarily reflect the views of the above sponsors.

Also issued as Operations Reserach Department Technical Report 94-4. Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribuition is unlimited.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

To my father, A.V.Ramana Murthy, and mother, Kalyani Devi.

Abstract

In this dissertation, we present some of the recent advances made in solving two-stage stochastic linear programming problems of large size and complexity. Decomposition and sampling are two fundamental components of techniques to solve stochastic optimization problems. We describe improvements to the current techniques in both these areas.

We studied different ways of using importance sampling techniques in the context of Stochastic programming, by varying the choice of approximation functions used in this method. We have concluded that approximating the recourse function by a computationally inexpensive piecewise-linear function is highly efficient. This reduced the problem from finding the mean of a computationally expensive function to finding that of a computationally inexpensive function. Then we implemented various variance reduction techniques to estimate the mean of a piecewise-linear function. This method achieved similar variance reductions in orders of magnitude less time than, when we directly applied variance-reduction techniques directly on the given problem.

In solving a stochastic linear program, the expected value problem is usually solved before a stochastic solution is attempted. This enables us to understand the value of the stochastic solution and also to speed-up the algorithm by making use of the information obtained from the solution of the expected value problem. We have devised a new decomposition scheme to improve the convergence of this algorithm. We have shown that this method gives a monotonically decreasing sequence of upper bounds whereas the regular decomposition scheme does not. We have also demonstrated that the added computational burden is insignificant compared to the reduction in the number of iterations and the resulting gain in CPU time.

Acknowledgements

Many people have helped me during my stay at Stanford. First and foremost, I would like to thank my advisor, Professor George B. Dantzig, for the guidance and support he has provided. I am particularly thankful to him for introducing me to the exciting area of Stochastic Programming. I am also grateful to have had the opportunity to be a Teaching assistant for him and gain insights into various aspects of Operations Research. I am very grateful to Dr. Gerd Infanger for all the valuable suggestions and constant words of encouragement that I received all through my program. I also thank him for allowing me the use of the software developed by him, without which I would not have had the opportunity to test many of the ideas in my research. I would like to thank Professor Donald Iglehart for his prompt and valuable comments regarding this dissertation.

I am thankful to Professor Samuel S. Chiu of the Engineering Economic Systems department and Professor Peter Glynn of the department of Operations Research, for serving on my orals committee. My thanks go to all the memebers of Operations Research department for providing me with an intelectually stimulating environment. I thank Gail Stein and Mako Bockholt for an adminstrative environment that is efficient and warm.

I am deeply indebted to my parents for encouraging me to take up this challenging endeavor and lending support that I could always depend on. I thank all my family and friends from India for the confidence they have shown in me and patience with my irregular communication. My sincere thanks go to all the friends who made my stay at Stanford enjoyable as well as educational. In particular, I would like to take this opportunity to thank my friends Costantinos Skiadas, Diane Erdmann.

Vijay Mehrotra, Nathan Chan, Gregeory DeCroix, Steve Murray, Jen-Chih Yao, Ramana Gollakotta, Sridhar Raman, Ennapadam Venkatraman, P.V.Krishnarao, Manickam Olagappan, Prasad Akella, Sripathi Acharya, Ashish Gupta, Nita Goyal, Caroline Stipe, Subbarao Gunupudi and Aparna Gunupudi.

Finally I thank my wife, Sita R. Kaluri, for the love and understanding she has shown that helped me to overcome many an obstacle.

Preface

Uncertainty has been one of the most difficult aspects to incorporate into an optimization model designed to solve complex real life problems. Often times, in practice, the uncertain parameters of the problem are replaced by their mean values or most probable values. While it was realised that a model that incorporates uncertainty is bound to give superior results to the one that does not, the resulting increase in the size and complexity of the models and non-existence of software that can solve such models quickly at a low cost has deterred modelers from abandoning the simplistic approach of the past.

The search for algorithms that solve these larger problems in less time continues to be a fundamental research goal. Advances in this area, together with the advances in the speed and capacity of computers in the 1980's and early 90's has made it possible to solve problems that were once considered intractable. In particular, it is now, within the scope of existing technology to solve models that hedge against some of the future uncertainties.

In this dissertation, we present some of the advances upto 1993 made in solving stochastic optimization problems. Decomposition and sampling are two of the fundamental components of these newer techniques for solving stochastic optimization problems. We describe improvements to the current techniques in both these areas, improvements that enable us to solve much larger problems in a shorter time.

We begin in Chapter I by describing the formulation of a two stage stochastic linear program, that can be used to model the uncertainty of many real-life problems. We also review a few approaches used to solve these problems.

Chapter II discusses the incorporation of monte carlo sampling into existing

algorithms to solve two-stage stochastic linear programs. Improvements to the existing methodology are suggested. The efficiency of these improvements are reported at the end of this chapter.

Chapter III illustrates the advances made in the area of decomposition techniques for non-stochastic problems. This is followed by numerical results.

We conclude with Chapter IV which summarizes the advances made in this dissertation and suggests future areas of research.

Table of Contents

Abstract iv

Acknowledgements v

Preface vii

I. Introduction 1

1.1. Two-stage Stochastic Linear Program . 1

1.2. Review of Solution Techniques 3

1.3. Current Methodology 4

II. Sampling Strategies 9

2.1. Introduction 9

2.2. Sampling methods 11

2.3. Importance Sampling 13

2.3.1. Additive Approximation 17

2.3.2. Need for other approximations 20

2.4. Improved additive approximation . . . 22

2.4.1. Finding a least cost scenario 23

2.4.2. Using any scenario as a base case 24

2.4.3. Exponential smoothing 25

2.5. Piecewise-linear approximation function 25

2.5.1. Accuracy of approximation 27

2.5.2. Ease of evaluation 31

2.5.3. Evaluating the Importance Density 31

2.5.4. Evaluation of the mean of approximation 35

2.6. Numerical Results 43

III. Decomposition methods 63

3.1. Introduction 63

3.2. Benders Decomposition 64

3.3. Primal-Dual Decomposition 68

3.4. Implementation Issues 75

Conclusion 78

Bibliography 80

Chapter I
Introduction

1.1. Two-stage Stochastic Linear Program

Planning under uncertainty has been a rapidly growing subject in the field of Operation Research. Since its inception in 1955 by Dantzig (1955) [8] and independently by Beale (1955) [3], many researchers have contributed to this area. Wets and Kall (1976) [26], Dempster (1980) [13], Ermoliev (1988) [16] and Infanger (1993) [24] are some of the books entirely devoted to this subject. Research in this area is started by formulating models that incorporate uncertainty. A typical stochastic programming problem can be formulated as:

$$\begin{aligned} &\text{Min } E(z(x)) \\ &\text{s.t. } x \in C \subset \Re^n \end{aligned} \tag{1.1}$$

where $z(x)$ is a random variable defined on the probability space $(\Omega, \mathcal{F}, P)$ and C is a convex set. (In the remainder of this thesis, we also use $E_P z(x,\omega)$ to mean the expected value of the random variable $z(x)$. In general, the expectation of a random variable X is represented either by $E(X)$ or by $E_P X(\omega)$). However, the exact nature of the constraints and the dynamic nature in which realizations of the random parameters are observed, give forth to different specific models. One such formulation is the classic two-stage stochastic linear program which is expressed as follows:

$$\begin{aligned} z^* = \min\ & cx + E_P Q(x,\omega) \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned} \tag{1.2}$$

where Q at any given decision x is a random variable defined on the probability space $(\Omega, \mathcal{F}, P)$. The value of Q at a given decision x and at a given point $\omega \in \Omega$ is denoted by $Q(x, \omega)$ and is given by the following expression:

$$\begin{aligned} Q(x,\omega) = \min\, & c_2(\omega)y(\omega) \\ \text{s.t. } & D(\omega)y(\omega) = d(\omega) + B(\omega)x \\ & y \geq 0 \end{aligned} \tag{1.3}$$

where c_2, d are random vectors and D, B are random matrices defined on the same probability space $(\Omega, \mathcal{F}, P)$.

In this formulation, decisions are made in two stages. In the first-stage, some decision x, an $n_1 \times 1$ vector, is made subject to $Ax = b, x \geq 0$ where A is a known $m_1 \times n_1$ matrix and b is a known $m_1 \times 1$ vector. The first-stage cost associated with this decision x is cx, where c is a known $1 \times n_1$ vector. After this, the specific outcome ω of the underlying random vector is observed. This determines the coefficients $c_2(\omega)$ which is a $1 \times n_2$ vector, $B(\omega)$ which is an $m_2 \times n_1$ matrix, $D(\omega)$ which is an $m_2 \times n_2$ matrix and $d(\omega)$ which is an $m_2 \times 1$ vector . Then the second-stage decision $y(\omega)$, an $n_2 \times 1$ vector, is made so as to minimize the cost $c_2(\omega)y(\omega)$ subject to the second-stage constraints $D(\omega) = d(\omega) + B(\omega)x$, $y \geq 0$. $Q(x, \omega)$ represents this minimum second-stage cost of the decision x at the outcome ω. It is also called the recourse function. The objective of the model is to choose a first-stage decision x, subject to the first-stage constraints, that minimizes the sum of first-stage cost and the expected second-stage costs.

As an example, consider the problem of capacity expansion for an electric utility company. Here the first stage decision would consist of the amount of new capacity to be acquired, the choice of technology used to generate electricity and placement of new facilities etc. The second stage decision would correspond to the

daily operations of all the plants operated by the company. The uncertainties can include demand for electricity, non-availability of built-in capacity due to random failures, etc. This problem can be modelled as a two-stage stochastic linear program described above.

1.2. Review of Solution Techniques

The two-stage stochastic linear program has been studied extensively. Solution techniques combine methods of Operations Research, like linear and non-linear programming, with traditional statistical methods, like estimation and monte-carlo sampling theory etc.

Some solution techniques are designed to take advantage of the special matrix structure of the problem. For example, Van Slyke and Wets (1969) [35] describe how Benders (1962) [4] decomposition can be applied to solve the two-stage stochastic linear programs.

Mathematical programming techniques are applied in special cases by Nazareth and Wets (1986) [30]. The same authors have given an overview of application of non-linear programming techniques in stochastic programming, see e.g., Nazareth and Wets (1988) [31]. However, in the case of stochastic optimization the exact function evaluations and gradients required by general non-linear programming techniques are difficult . To overcome this problem, stochastic quasigradient methods have been designed by Ermoliev (1988) [15]. Implementations of these methods has been studied by Gaivoronski (1988) [18].

When the exact solution of the problem is not possible or not practical to do, approximate schemes are devised to obtain bounds. This can efficiently be done in

the case where uncertainty is restricted to the right-hand side of the linear program in (1.3) . If so, the dual of (1.3) has a feasible region that is not random. Using this fact and Jensen's (1906) [25] inequality for convex functions, a lower bound for z^* can easily be constructed. Upper bounds can be constructed as explained in Edmundson (1956) [14] and Madansky (1959) [27].

Some of the techniques described above such as Van Slyke and Wets [35], consider all the scenarios of the underlying random parameters. This limits the application of such methods to problems with only a moderate number of scenarios. To tackle problems with large number of scenarios, sampling techniques have been combined with some of the methods mentioned above. For example, Gaivonronski and Nazareth (1989) [17] use a combination of generalized programming with sampling methods. Pereira et.al. (1989) [32] used the control variates technique along with Benders decomposition. Higle and Sen (1989) [22] developed a modified Benders Decomposition method that uses only one observation per iteration or a very small sample size. Dantzig and Glynn (1990) [7] have devised a combination of monte carlo importance sampling and decomposition techniques, which is further developed by them and Infanger (1991) [23]. In this thesis, we make significant improvements to their approach in both the areas of decomposition and sampling techniques.

In the remainder of this chapter, we will briefly describe this approach so as to lay the background for the enhancements we made. For more details on this approach and numerical results, we refer to Infanger (1993) [24].

1.3. Current Methodology

Let us consider a two-stage stochastic linear programming problem defined on a discrete probability space with finitely many scenarios. For the purpose of this discussion we will consider a model where randomness is restricted to the right-hand side of the feasible region in (1.3) , i.e., only B and d are random and c_2 and D are not random. Let $\Omega = \{\omega_1, \ldots, \omega_M\}$. We denote the probability of a scenario ω_i by p_i, $B(\omega_i)$ by B_i, $y(\omega_i)$ by y_i and $d(\omega_i)$ by d_i. Thus, the problem in (1.2) can be restated as a single linear program as follows:

$$
\begin{array}{lllllll}
z^* = \min & cx + p_1 c_2 y_1 + \ldots + p_M c_2 y_M & & & & & \\
\text{s.t.} & Ax & & & & & = b \\
 & -B_1 x + & Dy_1 & & & & = d_1 \\
 & \vdots & & \ddots & & & \vdots \\
 & -B_M x & & & + & Dy_M & = d_M \\
 & x\,, & y_1\,, & \ldots\,, & & y_M & \geq 0.
\end{array}
\tag{1.4}
$$

The current methodology adapted by Dantzig and Glynn (1990) [7] and Infanger [23] solves the problem in (1.4) by a combination of Benders Decomposition and importance sampling.

Benders decomposition algorithm is an iterative scheme in which the successively refining outer approximations are built for the expected value of the recourse function. Each iteration consists of a master problem and a set of subproblems. The master problem solves an approximation of the original problem, in which the recourse function is replaced by its current approximation. The subproblems are

used to refine the current approximation to the recourse function by evaluating the gradient of the recourse function at a given first-stage decision supplied by the master problem of that iteration.

The master problem at iteration t has the following form:

$$
\begin{aligned}
cx_t^* + \theta_t^* = \min\ (& \qquad cx + \theta) \\
\text{s.t.} \quad & Ax = b \\
& \sum_{i=1}^{M} p_i \pi^*(\omega_i, x_0^*) B_i x + \theta \geq \sum_{i=1}^{M} p_i \pi^*(\omega_i, x_0^*) d_i \\
& \vdots \\
& \sum_{i=1}^{M} p_i \pi^*(\omega_i, x_{t-1}^*) B_i x + \theta \geq \sum_{i=1}^{M} p_i \pi^*(\omega_i, x_{t-1}^*) d_i \\
& x \geq 0.
\end{aligned} \tag{1.5}
$$

At iteration $t = 0$, there will not be any constraints involving θ and the variable θ will be dropped from the problem. There are M subproblems at every iteration, one for each scenario ω of the underlying random parameter vector. The subproblem at iteration t, corresponding to scenario ω_i has the following form:

$$
\begin{aligned}
c_2 y^*(\omega_i, x_t^*) = \min\ & c_2 y_i \\
\text{s.t.}\ & Dy_i = d_i + B_i x_t^* \quad | \quad \pi^*(\omega_i, x_t^*) \\
& y_i \geq 0.
\end{aligned} \tag{1.6}
$$

The vectors $\pi^*(\omega_i, x_s^*)$, $i = 1, \ldots, M$ and $s = 0, \ldots, t-1$, used in the master problem (1.5) are the optimal dual multipliers of the subproblem corresponding to scenario ω_i in iteration s. Similarly, the x_t^* used in the right-hand side of the subproblem (1.6) is the optimal solution of the master problem at iteration t.

Observe that the master problem (1.5) is equivalent to the following problem:

$$
\begin{aligned}
z_t^t = \min\ & cx + Q^t(x) \\
\text{s.t.}\ & Ax = b \\
& x \geq 0
\end{aligned} \tag{1.7}
$$

where

$$Q^t(x) = \max_{0 \le s \le t-1} E_P(\pi^*(\omega, x_s^*)(d(\omega) + B(\omega)x)). \tag{1.8}$$

However, by writing the dual of (1.3) and by noting that $\pi^*(\omega, x_s^*)$ is a feasible solution for that problem, we can see that $Q^t(x) \le E_P Q(x, \omega)$. It then follows that, z_l^t is a lower bound for z^*.

Since x_t^* is a feasible first-stage decision, an upper bound for the optimal value z^* can be constructed as follows:

$$z_u^t = cx_t^* + \sum_{i=1}^{M} c_2 y^*(\omega_i, x_t^*). \tag{1.9}$$

The flow of information in this algorithm is shown in the following figure.

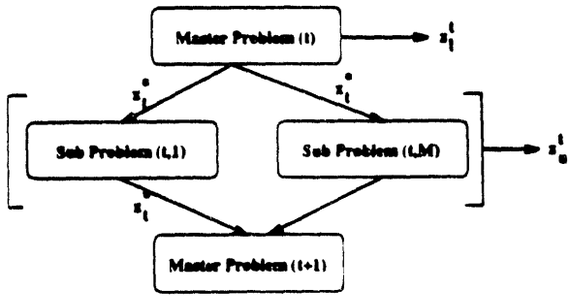


Fig.1 : Flow of information in Benders Decomposition

According to the approach just described, one would have to solve M subproblems at every iteration. This can be prohibitively expensive if M is large, as it is likely to be in complex real-life models. Therefore a promising approach is to modify the algorithm so that only a few subproblems have to be solved in each iteration. These can be selected through a statistical sampling scheme which enables us to make probabilistics statements about how close the objective of the resulting

solution is to the true optimum value. Thus, if we choose S_t to be the set of samples chosen in the t-th iteration, we would have a master problem which has the form:

$$\begin{aligned} cx_t^* + \theta_t^* = \ & \min cx + \theta \\ & \text{s.t. } Ax \quad = b \\ -\tfrac{1}{|S_0|} \sum_{\omega \in S_0} \pi^*(\omega, x_0^*) B(\omega) x + \theta & \geq \tfrac{1}{|S_0|} \sum_{\omega \in S_0} \pi^*(\omega, x_0^*) d(\omega) \\ -\tfrac{1}{|S_{t-1}|} \sum_{\omega \in S_{t-1}} \pi^*(\omega, x_{t-1}^*) B(\omega) x + \theta & \geq \tfrac{1}{|S_{t-1}|} \sum_{\omega \in S_{t-1}} \pi^*(\omega, x_{t-1}^*) d(\omega) \\ x \quad & \geq 0 \end{aligned} \tag{1.10}$$

and $n_t = |S_t|$ subproblems defined by (1.6) . We would then have

$$\bar{z}_u^t = cx_t^* + \sum_{\iota \in S_t} c_2 y^*(\omega_\iota, x_t^*) \tag{1.11}$$

which is an unbiased estimator of an upper bound of z^*. However, $\bar{z}_l^t = cx_t^* + \theta_t^*$ is not necessarily an unbiased estimator of a lower bound for z^*. Infanger (1993) [23] explains how to modify the lower bound in order to construct a confidence interval for z^*. The choice of sampling scheme used is crucial for fast convergence of this algorithm. In their approach, Dantzig and Glynn (1990) [7] and Infanger (1993) [23] used importance sampling with great success. We provide a detailed description of their scheme in Chapter 2.

This basic framework for solving stochastic programs has been further developed in three directions. Morton (1993) [28] studied the stopping rules for algorithms with sampled upper and lower bounds and applied them to solve stochastic programs. Infanger (1993) [24] and Morton (1993) [28] studied extenstions of this framework to Multi-stage stochastic programs. In this dissertation we study modifications to the sampling schemes and decomposition methods in order to improve the speed of the algorithm.

Chapter II
Sampling Strategies

2.1. Introduction

Multi-dimensional integration is an important aspect of stochastic programming. In general, whenever one has to make an expected-value statement about a model that incorporates uncertainty, one is faced with the problem of integration. When one incorporates many random parameters into a model in order to capture the nature of uncertainty, each additional random parameter contributes to higher dimensionality of the problem. When the parameters are discrete this leads to the explosion of the number of possible scenarios. Moreover, in stochastic programming the functions that are to be integrated are rarely smooth and analytical. Thus, higher dimensionality and the non-smooth nature of the integrand mainly contribute to the challenges of multi-dimensional integration.

Multi-dimensional integration has been studied widely. Some of the survey papers are, e.g, Deak (1988) [12], Haber (1970) [21] and books , e.g., Stroud (1971) [34] and Davis and Rabinowitz (1975) [11]. In particular, the first paper studies multi-dimensional integration in the context of stochastic programming.

We begin our discussion by showing where multi-dimensional integration arises in the context of stochastic programming. This enables us to exhibit their special structure. We follow this with a brief discussion of variance-reduction techniques used to improve the accuracy of the multi-dimensional integration. We then present

importance sampling as a tool to address integration problems in the context of stochastic programming. Next we describe the current usage of importance sampling and a scope for its improvement. Finally, we suggest a new way of implementing importance sampling and show its effectiveness through numerical examples.

Multi-dimensional integrals arise naturally in stochastic programs. They are part of the models as well as the algorithms that are designed to solve the models. In the two-stage stochastic programming model, described in (1.2) , the second stage costs are described as an integral. Even if we decide to solve this problem through a deterministic approach by using the expected values of the inherent random parameters, to understand the effectiveness of such a method we still would need to perform a multi-dimensional integration. Clearly, in algorithms such as the ones described in the previous chapter, which attempt to hedge against the uncertainty, multi-dimensional integration has to be performed at every iteration. Also in chance constrained models described in Prékopa (1988) [33], one often needs to compute the probability with which a given constraint is satisfied. This too falls under multi-dimensional integration. In the ensuing discussion, we treat integration and evaluation of expectation synonymously.

There are many different techniques to carry out numerical integration. For a review of these techniques refer to Deak [12]. Most methods except those based on the concept of statistical sampling are not practical for large problems, especially those problems having non-smooth functions or non-smooth probability distributions. Since, the problems we consider fall into this latter category, we will restrict ourselves to sampling methods.

2.2. Sampling methods

These are used for integrations that are too difficult to solve analytically. In general, as the number of variables increases, sampling becomes the only viable choice to evaluate expectations of multi-dimensional functions. The fundamental idea behind a sampling method is to approximate the given probability distribution by an empirical distribution having nice properties and then use it to approximate the integral in question. Consider a random variable f defined on a probability space (Ω, F, P). For the purpose of illustration we will assume the existence of a density function p. Consider the problem of estimating θ, the mean of f, given by,

$$\theta = \int_{\Omega} f(\omega)p(\omega)d\omega.$$

Let $\omega_1, \ldots, \omega_n$ be n independently randomly drawn samples according to the density p. Let $\hat{p}$ be the approximation to the density p given by

$$\hat{p}(\omega) = \begin{cases} \frac{1}{n} & \text{if } \omega \in \{\omega_1 \ldots \omega_n\} \\ 0 & \text{otherwise} \end{cases}. \tag{2.1}$$

The integral using this approximate probability mass function is given by

$$\hat{\theta} = \sum_{\Omega} f(\omega)\hat{p}(\omega) = \frac{1}{n}\sum_{i=1}^{n} f(\omega_i). \tag{2.2}$$

$\hat{\theta}$ is an estimator of θ. There are many nice properties of such an estimator, like unbiasedness, which make it a very popular method to approximate integrals. The variance of this estimator is given by

$$\frac{\sigma^2}{n} = \frac{1}{n}\left\{\int_{\Omega} f^2(\omega)p(\omega)d\omega \; - \; \theta^2\right\}. \tag{2.3}$$

An unbiased estimate of σ^2/n is

$$\frac{S^2}{n}, \tag{2.4}$$

where

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(f(\omega_i) - \hat{\theta})^2. \tag{2.5}$$

This variance can be quite large and in such a case we need a large sample size to get the desired accuracy. The convergence rate of this estimator is $1/\sqrt{n}$. For example if a sample of size n_0 gives us an estimator accurate to one decimal point, it requires a sample of size $100n_0$ using this method to give an estimator accurate to two decimal places.

Clearly, this is unsatisfactory in many cases. Especially, in stochastic programming, where expectations have to be evaluated at every iteration of an algorithm, large sample sizes can slow down the algorithm considerably. However, sampling need not be done naively, i.e., simply following the given probability distribution as described above. It can be done instead in conjunction with special techniques, to either improve accuracy or speed up convergence. One such class of special techniques is known as variance-reduction techniques. Notable among these are importance sampling, control cariates technique, antithetic variates technique, common random-numbers technique and Stratification. These techniques help to reduce the variance of the estimator in different ways.

For example, suppose we are interested in estimating the expectation of the difference of two random variables, which are positively correlated. Recall that, to generate a sample from any probability distribution, we first generate a sequence of random numbers uniformly in the interval $[0, 1]$ and apply a suitable transformation on these. The common random-numbers technique uses the same sequence

of random numbers, instead of estimating both the parameters independently, to reduce the variance. On the other hand, if the original random variables are negatively correlated, we can use one stream of random numbers, say $U_1, \ldots, U_n$, to generate sample points of one random variable and use $1 - U_1, \ldots, 1 - U_n$ for the other random variable where $0 \leq U_i \leq 1$.

On the other hand, suppose we want to estimate the expectation of one random variable, by taking average values over a large number of replications. The method of antithetic variates suggests creating dependence between replications in order to reduce the variance of the estimator. Method of stratification refers to dividing the sample space into disjoint 'strata' and taking samples from each stratum separately. This would enable us to sample more from a stratum that contributes more to the overall variance. A detailed discussion of these methods is available in Bratley et.al. (1983) [6]. However, in stochastic programming, we do not often have enough information on the monotonicity of the functions involved or the variability in different strata to apply these techniques. Thus we decided to use a combination of importance sampling and control variate techniques as a variance-reduction tool.

In the future, we will refer to the scheme, described in (2.1) -(2.5) , where no effort was made to reduce variance, as *naive sampling*. We define the *merit* of any variance-reduction technique, in relation to that of the naive-sampling approach.

2.3. Importance Sampling

Importance sampling is a variance-reduction tool that is often used in simulation of complex phenomena. It is particularly designed to address the situation in which there are certain scenarios that are extremely rare, yet significantly

contribute to the average measure of performance that is being estimated. Naive sampling would take a long time before some of these rare events are picked up in the sample. An estimate computed without observing these rare events might not accurately represent the true value of the desired parameter. These considerations usually translate into large variances of the estimator. In trying to rectify this situation, importance sampling method tries to reassign the probabilities to the events so that the new probability of an event represents both its *natural frequency of occurrence* and its *importance*. The importance of an event is assessed by the contribution of it to the parameter that is being estimated. Of course, a certain corrective action has to be taken to eliminate the bias introduced through this reassignment. For more details on importance sampling we refer to Bratley et.al [6], and Glynn and Iglehart (1989) [20].

First we formalize these ideas below. Then we describe the rare events that are encountered in stochastic programming. We go on to describe why importance sampling is a suitable tool even in cases where there are no special very rare events. This discussion is followed by various implementation issues of importance sampling.

Let

$$\theta = \int_{\Omega} f(\omega)p(\omega)d\omega$$

where the outcome ω of the underlying random parameter is a vector with r components $(\omega^1, \ldots, \omega^r)$. In naive sampling we choose a set of samples $\{(\omega_1, \ldots, \omega_n)\}$ according to the multivariate probability density p. At each of these points $\omega = \omega_i$ we evaluate the function $f(\omega)$. In importance sampling , we use a new probability density $q(\omega)$ to choose sample points. However, if we evaluate the function $f(\omega)$ at these points, we would be estimating

$$\int_{\Omega} f(\omega)q(\omega)d\omega$$

which is not the value θ we want. Therefore, as a corrective action, we evaluate, with respect to the probability density $q(\omega)$, the expected value of

$$f(\omega)p(\omega)/q(\omega).$$

Observe that, this restricts the choice of the new probability density to those densities that are positive for every $\omega \in \Omega$. Thus we are estimating

$$\int_\Omega \frac{f(\omega)p(\omega)}{q(\omega)} q(\omega)d\omega$$

which also equals θ, the desired parameter. The variance of this estimator is given by

$$\begin{aligned} \sigma_i^2/n &= \frac{1}{n}\left\{\int_\Omega \frac{f^2(\omega)p^2(\omega)}{q^2(\omega)} q(\omega)d\omega - \theta^2\right\} \\ &= \frac{1}{n}\left\{\int_\Omega \frac{f^2(\omega)p(\omega)}{q(\omega)} p(\omega)d\omega - \theta^2\right\}. \end{aligned} \tag{2.6}$$

Clearly, the amount of variance reduction, if any, depends on the choice of $q(\omega)$. For example, as we said earlier, if $q(\omega)$ is chosen to be proportional to $f(\omega)p(\omega)$, then it must equal $f(\omega)p(\omega)/\theta$. This is because, in order for $q(\omega)$ to be a probability density, it has to integrate to one. Using this choice of $q(\omega)$ we will get a variance of zero. However, this choice assumes we know the value of θ, the very parameter which we are trying to estimate, and hence is not available to us as a correction factor. However, it does provide us with a clue for finding a new density $q(\omega)$ that is approximately proportional to the product $p(\omega) \times f(\omega)$ and at the same time is easy to use. In the future we refer to such a $q(\omega)$ as an ***importance density***. We will discuss how to achieve this end after we motivate the use of importance sampling in the context of stochastic programming.

To construct a good importance density, we will look into both the function $f(\omega)$ and the probability density $p(\omega)$ and try to approximate them. We denote the

approximation function for $f(\omega)$ by $\Gamma(\omega)$ and we denote the approximation function for $p(\omega)$ by $\eta(\omega)$. Thus the importance density $q(\omega)$ will be given by

$$\frac{\Gamma(\omega)\eta(\omega)}{\bar{\Gamma}} \tag{2.7}$$

where $\bar{\Gamma} = \int_\Omega \Gamma(\omega)\eta(\omega)d\omega$. In estimating the expected second stage costs of a stochastic program, the variation in the function $f(\omega)$ typically plays a larger role than the variation in the probability of different scenarios. Therefore we will first concentrate on only approximating the function $f(\omega)$ in order to construct an importance density. Thus we have

$$\theta = \bar{\Gamma} \int_\Omega \rho(\omega)q(\omega)d\omega \tag{2.8}$$

where, $\bar{\Gamma} = \int_\Omega \Gamma(\omega)p(\omega)d\omega$, $q(\omega) = \Gamma(\omega)p(\omega)/\bar{\Gamma}$ and

$$\rho(\omega) = \frac{f(\omega)}{\Gamma(\omega)}. \tag{2.9}$$

Let us recall that the function that we are trying to integrate is given by

$$\begin{aligned} f(\omega) = \min\, & c_2^T y \\ \text{s.t.}\quad & Dy = d(\omega) + B(\omega)x \\ & y \geq 0 \end{aligned} \tag{2.10}$$

where x is a known vector. By writing the dual of the above linear program, we can also express it as

$$\begin{aligned} f(\omega) = \max\, & (d(\omega) + B(\omega)x)\pi_2^T \\ \text{s.t.}\quad & D^T\pi_2 \leq c_2\,. \end{aligned} \tag{2.11}$$

Often, the random vector d and the random matrix B have a simple additive form in terms of the components of the underlying random parameter vector. For example,

$$d(\omega) = d_0 + d_1\omega^1 + \ldots + d_{r_d}\omega^{r_d} \tag{2.12}$$

$$B(\omega) = B_0 + B_1\omega^{r_d+1} + \ldots + B_{r-r_d}\omega^r \tag{2.13}$$

where $d_0, \ldots, d_{r_d}$ are deterministic vectors and $B_0, \ldots, B_{r-r_d}$ are deterministic matrices and $\omega^1, \ldots, \omega^r$ are scalar components of the outcome ω. This structure is particularly useful if the components of the random parameter vector are independently distributed, i.e., $p(\omega) = \Pi_{i=1}^{r} p_i(\omega^i)$.

2.3.1. Additive Approximation

To estimate the expected value of this function, R.Entriken and M.Nakayama in Dantzig et.al [10]developed an approximation function that is additive in the components of the underlying random parameter vector. This function has the form:

$$\Gamma(\omega) = \Gamma_0 + \Gamma_1(\omega^1) + \ldots \Gamma_r(\omega^r). \tag{2.14}$$

To choose the constant Γ_0 and the univariate functions $\Gamma_1, \ldots, \Gamma_r$ they considered what is called a base case scenario Ψ. The approximation is designed to be an additive function that equals the original function $f(\omega)$ at all points obtained by varying one coordinate from the base case scenario while keeping the others fixed at the base case values. This definition leads us to the function:

$$\begin{aligned}
\Gamma(\omega) = & f(\psi^1, \ldots, \psi^r) \\
& + f(\omega^1, \ldots, \psi^i, \ldots, \psi^r) - f(\psi^1, \ldots, \psi^r) \\
& \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
& + f(\psi^1, \ldots, \omega^i, \ldots, \psi^r) - f(\psi^1, \ldots, \psi^r) \\
& \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
& + f(\psi^1, \ldots, \psi^i, \ldots, \omega^r) - f(\psi^1, \ldots, \psi^r).
\end{aligned} \tag{2.15}$$

Clearly, we have

$$\Gamma(\psi^1, \ldots, \omega^i, \ldots, \psi^r) = f(\psi^1, \ldots, \omega^i, \ldots, \psi^r)$$

for every $1 \leq i \leq r$ and for every ω^i. Moreover, the univariate function Γ_i in (2.14) is given by

$$\Gamma_i(\omega^i) = f(\psi^1, \ldots, \omega^i, \ldots, \psi^r) - f(\psi^1, \ldots, \psi^r) \tag{2.16}$$

and the scalar Γ_0 is given by

$$\Gamma_0 = f(\psi^1, \ldots, \psi^r). \tag{2.17}$$

In this case, application of importance sampling can be described as follows.

$$\begin{aligned}
\theta &= \int_\Omega f(\omega)p(\omega)d\omega \\
&= f(\Psi) + \int_\Omega \frac{f(\omega) - f(\Psi)}{\Gamma(\omega) - \Gamma(\Psi)} (\Gamma(\omega) - \Gamma(\Psi))\, p(\omega)d\omega \\
&= f(\Psi) + \int_\Omega \frac{f(\omega) - f(\Psi)}{\Gamma(\omega) - \Gamma(\Psi)} \left(\sum_{i=1}^{r} \Gamma_i(\omega^i) \right) p(\omega)d\omega \\
&= f(\Psi) + \sum_{i=1}^{r} \bar{\Gamma}_i \int_\Omega \frac{f(\omega) - f(\Psi)}{\Gamma(\omega) - \Gamma(\Psi)} \frac{\Gamma_i(\omega^i)p(\omega)}{\bar{\Gamma}_i} d\omega \\
&= f(\Psi) + \sum_{i=1}^{r} \bar{\Gamma}_i \int_\Omega \rho(\omega) \frac{\Gamma_i(\omega^i)p(\omega)}{\bar{\Gamma}_i} d\omega
\end{aligned} \tag{2.18}$$

where $\rho(\omega) = \frac{f(\omega)-f(\Psi)}{\Gamma(\omega)-\Gamma(\Psi)}$ and $\bar{\Gamma}_i = \int_\Omega \Gamma_i(\omega)p(\omega)d\omega$

When the components of the underlying random parameter vector are independently distributed, we have

$$\begin{aligned}
\theta &= f(\Psi) + \sum_{i=1}^{r} \bar{\Gamma}_i \int_\Omega \rho(\omega) \frac{\Gamma_i(\omega^i) \prod_{j=1}^{r} p_j(\omega^j)}{\bar{\Gamma}_i} d\omega \\
&= f(\Psi) + \sum_{i=1}^{r} \bar{\Gamma}_i \int_\Omega \rho(\omega) q_i(\omega^i) \prod_{\substack{j=1 \\ j \neq i}}^{r} p_j(\omega^j)
\end{aligned} \tag{2.19}$$

where

$$q_i(\omega^i) = \frac{\Gamma_i(\omega^i)p_i(\omega^i)}{\bar{\Gamma}_i}. \tag{2.20}$$

Notice that, in this case of independent distributions, evaluating $\bar{\Gamma}_i$ is reduced to a one dimensional integration. To evaluate this exactly, we need to be able

to evaluate the function $\Gamma_i(\omega^i)$ at all possible outcomes ω^i. In practice, when the random parameters are discrete, number of outcomes of any single random parameter are not too high. Therefore, $\bar{\Gamma}_i$ can be computed easily.

Thus, to implement importance sampling in this case, first we perform r one dimensional integrals to evaluate $\bar{\Gamma}_i$'s. Then we compute r modified one dimensional densities $q_i(\omega^i)$. The original parameter θ is expressed as a sum of r integrals, each of which is estimated through sampling. We select a sample size n_i to estimate the i-th integral. The n_i's are chosen to add up to a fixed sample size n allocated to estimate θ and are chosen roughly proportional to $\bar{\Gamma}_i$.

To estimate the i-th integral, we choose a set of sample vectors $\{\omega_{i1}, \ldots, \omega_{in_i}\}$ where the i-th component of each vector is chosen according to the new probability density q_i and all the other components are chosen with respect to the original probability densities. Let

$$\hat{M}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \rho(\omega_{ij}) \tag{2.21}$$

and

$$\hat{S}_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (\rho(\omega_{ij}) - M_i)^2. \tag{2.22}$$

The estimator for θ is given by

$$\hat{\theta} = f(\Psi) + \sum_{i=1}^{r} \bar{\Gamma}_i \hat{M}_i \tag{2.23}$$

with estimated variance of

$$\hat{\sigma}_\theta^2 = \sum_{i=1}^{r} \frac{\bar{\Gamma}_i^2 S_i^2}{n_i}. \tag{2.24}$$

By choosing an appropriate base case, the variance in (2.24) can be much smaller than the variance in (2.4) . The experiment of M.Nakayama in Dantzig

et.al. [10]consisted of estimating the second stage costs of a multi-area capacity expansion model with 192 universe scenarios. They used a sample size of 10 for both importance sampling and naive sampling and achieved a variance-reduction of 20000 times.

2.3.2. Need for other approximations

The additive approximation described in the previous section works very well in certain problems. In the capacity expansion example mentioned above, the second stage costs expressed as a function of the random scenario are almost additive, thereby giving us a very high degree of variance-reduction. This would be the case if, for various levels of demand contingencies, the method of supplying remains the same. For example, if the capacity built-in is very high, then all demand contingencies can be met by simply operating the plants at appropriate levels. On the other hand, if the capacity is very low, most of the time the demand is met through an outside source with a different, and usually very much higher, cost. In either case, the cost function would remain more or less additive. Mathematically speaking, if the optimal basis in (2.11) remains the same for most values of ω, then the function would be very close to being additive in ω. However, there will be cases in which the built-in capacity is enough to meet the demand up to a certain level but not beyond that. In such cases, the cost function would not be additive and the additive approximation would not give very good results, though it still might be a lot better than naive sampling.

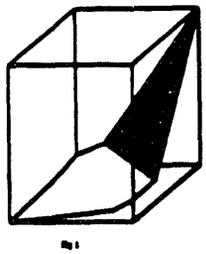


Fig. 1

The picture in Fig. 1 illustrates this point. In this example, the cost function is made up of two planes of different slopes. By taking the base case to be the scenario corresponding to the lower left hand corner of the box in Fig. 1, the additive approximation function is exactly equal to the true function along the two axes. Since all the points on the two axes have the same slope, the resulting additive approximation is a additive linear function which simply projects the lower slope onto the complete domain. It is easy to see that this would underestimate the real function considerably in the neighborhood of the upper diagonal corner D.

The performance of additive approximation depends on the choice of the base case Ψ. For example, by taking the upper right hand side corner in Fig. 1 as the base case, one can get a better approximation. In general, for the the new density $q_i(\omega^i)$ in (2.20) to be non-negative we must have $\Gamma_i(\omega^i)$ in (2.16) to be always non-negative or always non-positive. This is easily achieved if we can identify a scenario under which the second stage costs are minimized or maximized. For example, if the random parameters involve exogenous demands and levels of built-in capacity that are actually available, choosing lowest possible demands and highest levels of available capacities gives us a scenario which minimizes the second stage costs. However, when no such scenario is naturally available or when finding such a scenario is computationally expensive we need a different strategy. We will address this problem in later sections.

Finally, additive approximation by definition ignores the impact of joint variability of the arguments on the function. Therefore, it can be conjectured that the higher the number of arguments, the greater the chance that additive approximation does not provide a good approximation to the original function. We will demonstrate through numerical examples where this indeed is the case.

For the above mentioned reasons we have decided to look at other approximation functions to implement importance sampling. First, let us identify some of the important characteristics that a 'good' approximation function should posses. These are described below.

- $\Gamma(\omega)$ should be a good approximation to $f(\omega)$
- It should be easy to evaluate $\Gamma(\omega)$

(2.25)

- It should be easy to sample with the new density $q(\omega)$ obtained from $\Gamma(\omega)$.
- It should be easy to compute the mean of the approximation function, $\bar{\Gamma}$.

We can evaluate different schemes that we mentioned so far, under the above criteria. Observe that naive sampling can be considered as an importance sampling scheme in which a constant function is used as the approximation function. Therefore, it satisfies the last three of the above criteria but fails in the first one. Using the function $f(\omega)$ itself as an approximation would do exactly the opposite. It would satisfy the first criterion very well but fails in the last three. Additive approximation satisfies the last three criteria, except when finding a proper base case becomes computationally expensive. However, it satisfies the first criterion only in some cases. Thus we need to explore other ways of implementing importance sampling.

2.4. Improved additive approximation

Before we study other approximation functions, we first tried to modify the additive approximation function. This was done in mainly two different areas.

First we studied the effects of using different base cases. We looked into ways of implementing additive approximation scheme as is, by identifying a scenario that would have least cost among all scenarios obtained by changing one coordinate of the base case. Then we attempted to change the implementation to enable us to use any scenario as a base case, by taking care of the positive and negative parts of the approximation function separately. Finally, we tried an exponential smoothing scheme to enhance the performance of additive approximation. We describe these attempts in the following sections.

2.4.1. Finding a least cost scenario

One way to ensure that the importance density $q_i(\omega^i)$ in (2.20) is non-negative, is to find a base case Ψ which is an argmin of the function $f(\omega)$. Recall the form of $d(\omega)$ and $B(\omega)$ expressed in (2.12) and in (2.13) . Also, assume that ω^i has a compact support with a minimum value of ω_l^i and a maximum value of ω_u^i. Then, the ω component of the solution of the following linear program gives us a scenario Ψ which minimizes $f(\omega)$.

$$\begin{aligned} &\min c_2^T y \\ &\text{s.t } Dy - \sum_{i=1}^{r_d} d_i\omega^i - \sum_{i=r_d+1}^{r} B_{i-r_d} x\omega^i = d_0 + B_0 x \\ &\quad \omega_l^i \le \omega^i \le \omega_u^i, \ \forall 1 \le i \le r \\ &\quad y \ge 0 \end{aligned} \tag{2.26}$$

If the support of ω^i is not the entire interval $[\omega_l^i, \omega_u^i]$, for example if ω^i is discrete, then we need to make sure that $p(\Psi)$ is positive. Towards that end, let us suppose that the problem (1.2) is of complete recourse, i.e., problems in which the linear program in (2.10) is feasible for every ω. In that case the above problem becomes

$$\begin{aligned} &\min f(\omega) \\ &\text{s.t. } \omega_l^i \le \omega^i \le \omega_u^i, \ \forall 1 \le i \le r \end{aligned}$$

and $f(\omega)$ is a proper concave function on the domain. Therefore, a minimum occurs at an extreme point of the domain which gives us a scenario where each ω^i is either equals ω_l^i or equals ω_u^i. This gives us a scenario where $p(\Psi)$ is positive. On the other hand, if we do not have full recourse, $f(\omega)$ may not be finite on the entire domain and we cannot prove that an optimal solution occurs at an extreme point.

2.4.2. Using any scenario as a base case

If we can not find a least cost scenario easily, we can modify the additive approximation function as follows to make the importance density $q_i(\omega^i)$ in (2.20) non-negative.

$$\begin{aligned}\Gamma(\omega) = & f(\psi^1, \ldots, \psi^r) \\ & + f(\omega^1, \ldots, \psi^i, \ldots, \psi^r) - l_1 \\ & \vdots \qquad \vdots \qquad \vdots \\ & + f(\psi^1, \ldots, \omega^i, \ldots, \psi^r) - l_i \\ & \vdots \qquad \vdots \qquad \vdots \\ & + f(\psi^1, \ldots, \psi^i, \ldots, \omega^r) - l_r \end{aligned} \tag{2.27}$$

where

$$l_i = \min_{\omega^i} f(\psi^1, \ldots, \omega^i, \ldots, \psi^r). \tag{2.28}$$

This ensures us that

$$\begin{aligned} q_i(\omega^i) &= \frac{\Gamma_i(\omega^i)p_i(\omega^i)}{\bar{\Gamma}_i} \\ &= \frac{f(\psi^1, \ldots, \omega^i, \ldots, \psi^r) - l_i}{\int_\Omega (f(\psi^1, \ldots, \omega^i, \ldots, \psi^r) - l_i)p(\omega)d(\omega)} \\ &\geq 0. \end{aligned} \tag{2.29}$$

However, with this modification we lose the property that the approximation function and the real function agree on the axes projected from the base case. But

the ability to consider any choice of ω as a base case gives us an opportunity to experiment with a few different cases and select a good one.

2.4.3. Exponential smoothing

The advantage of using an additive approximation is the ability to express the original integral as a sum of a number of integrals, in each of which the probability function retains the product form. Therefore, when an additive approximation function does not perform very well, it is possible to tune it, without losing the advantage of additivity. Consider the following function,

$$\Gamma(\omega) = \Gamma_0 + e^{\alpha_1(\omega^1-\psi_1)}\Gamma_1(\omega^1) + \ldots e^{\alpha_i(\omega^i-\psi_i)}\Gamma_i(\omega^i) + \ldots e^{\alpha_r(\omega^r-\psi_r)}\Gamma_r(\omega^r) \quad (2.30)$$

where $\Gamma_i(\omega^i)$ are as defined in (2.16) . By controlling the parameters $\alpha_1, \ldots, \alpha_r$ we can control the ratio $\rho(\omega) = \frac{f(\omega)}{\Gamma(\omega)}$ more efficiently. Note that the only difference in the implementation of the importance sampling scheme is in the importance density which is given by

$$q_i(\omega^i) = \frac{e^{\alpha_i(\omega^i-\psi_i)}\Gamma_i(\omega^i)}{\int e^{\alpha_i(\omega^i-\psi_i)}\Gamma_i(\omega^i)p_i(\omega^i)d\omega^i}.$$

Therefore, it still involves only one dimensional integrals to compute the importance density. The only extra work is in choosing the parameters α_i.

2.5. Piecewise-linear approximation function

Since the additive approximation may or may not perform adequately in a specific problem, we want to design an approximation function that works well in a

larger number of problems. Towards that end, we will propose and demonstrate the use of a piecewise-linear function as an approximation function for use in importance sampling. Consider the form of the function $f(\omega)$ as expressed in (2.11) . If the feasible region of this linear program is empty, it would mean the second stage costs are unbounded below. Thus, the problem (1.2) is unbounded below. On the other hand, if the linear program in (2.11) is unbounded above, it would mean, for some ω the second stage costs in (1.3) are unbounded, there by making the parameter θ, which in this case is $E(Q(x,\omega))$ in (1.3) , to be infinity. Therefore, the only interesting case that needs to be addressed is, the case in which the second stage costs are finite for each ω. This allows us to assume that the feasible region in (2.11) is non-empty and to restrict our attention to its extreme points and ignore the extreme rays if any.

Hence, we can express the function $f(\omega)$ as follows.

$$f(\omega) = \max_{1 \leq i \leq L} \pi_2^i(d(\omega) + B(\omega)x) \tag{2.31}$$

where $\pi_2^i, 1 \leq i \leq L$ are all the extreme points of the dual feasible region in (2.11) . Considering the form of this function, it is natural to try the following function as an approximation function.

$$\Gamma(\omega) = \max_{1 \leq i \leq l} \pi_2^{j_i}(d(\omega) + B(\omega)x) \tag{2.32}$$

where $j_1, \ldots, j_l$ are a sequence of distinct integers between 1 and L. Approximation function of this form is also used by Pereira et.al. (1989) [32] in conjunction with a control variates technique. As the following picture demonstrates, often piecewise-linear function with only few *pieces* can give a very good approximation to the original function.

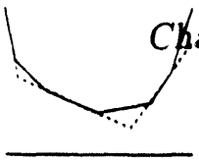


Fig 2

To asses the performance of such an approximation function in an importance sampling procedure let us look at the criteria defined in (2.25) .

2.5.1. Accuracy of approximation

It is clear that if we choose $l = L$ then $\Gamma(\omega) \equiv c(\omega)$. However from a computational point of view, this observation is of no use, since generating all the extreme points of a polyhedron and checking that we have all of them is impractical in large problems because L is too large in practice, or so it would appear. On the otherhand, it often turns out in practice that L is not too large, and so it is reasonable to expect that a small l will suffice to obtain a very good approximation of the polyhedral set.

Therefore, it is worthwhile to investigate how many dual extreme points do we expect to see in practice. Repeated experiments have shown that this number can be surprisingly small compared to what we might expect from the large number of 'possible' combinations that could result from the dimensions of the problem. For example, in a linear program with coefficient matrix of dimensions 4×10 there can be *at most* $\binom{10}{4} = 210$ possible feasible dual bases. However in randomly generated problems, there will be on the average only four dual extreme points! This claim is substantiated by the following result.

Theorem: Consider the following linear program

$$\begin{aligned} &\min c^T x \\ &\text{s.t. } Ax = b \\ &\qquad x \geq 0 \end{aligned}$$

where A is an $m \times n$ matrix. Suppose all the elements of A and c are independently and identically distributed with uniform distribution on the interval $[-1\ 1]$. Then the expected number of dual extreme points for this linear program equals $\binom{n}{m}\frac{1}{2^{n-m}}$.

Proof: Note that there are $\binom{n}{m}$ different square matrices in A that we need to consider as possible dual feasible bases. Let us denote them as B_i for $i = 1, \ldots, \binom{n}{m}$ and define N_i as the remaining part of the matrix A. Observe that each of the matrices B_i is non-singular with probability one, as the probability that the continuous random variable $\det(B_i)$ assumes the value zero, equals zero. Let T be the total number of dual feasible bases. The value of T is given by:

$$T = \#\left\{B_i : c_{N_i} - c_{B_i} B_i^{-1} N_i \geq 0\right\}.$$

Let $I(C)$ be the indicator function that assumes value 1 if the condition C is true and 0 otherwise. Since expectation is an additive operator the expected value of T is given by:

$$E(T) = \sum_{i=1}^{\binom{n}{m}} E(I(c_{N_i} - c_{B_i} B_i^{-1} N_i \geq 0)).$$

Without loss of generality, let us assume that B_1 consists of the first m columns of A. Since for each i the joint distribution of $(B_i, N_i, c_{B_i}, c_{N_i})$ is the same, we have

$$\begin{aligned} E(T) &= \binom{n}{m} E(I(c_{N_1} - c_{B_1} B_1^{-1} N_1 \geq 0)) \\ &= \binom{n}{m} E_{c_{B_1} B_1^{-1}}(E(I(c_{N_1} - c_{B_1} B_1^{-1} N_1 \geq 0 \,|\, c_{B_1} B_1^{-1}))) \\ &= \binom{n}{m} E_{c_{B_1} B_1^{-1}}(E(\prod_{j=m+1}^{n} I(c_j \geq c_{B_1} B_1^{-1} A_{.j} \,|\, c_{B_1} B_1^{-1}))). \end{aligned}$$

For $j = m+1, \ldots, n$ the conditional distributions of $(c_j, A_{.j})$ are identical and independent, given $c_{B_1} B_1^{-1}$. Therefore

$$E(T) = \binom{n}{m} E_{c_{B_1} B_1^{-1}}(E(I(c_{m+1} - c_{B_1} B_1^{-1} A_{.m+1} \geq 0 \,|\, c_{B_1} B_1^{-1})))^{n-m}.$$

However, c_{m+1} is a symmetric random variable and given $c_{B_1}B_1^{-1}$ we also have $c_{B_1}B_1^{-1}A_{.m+1}$ symmetric, as linear combinations of symmetric random variables is symmetric. Therefore their difference $c_{m+1} - c_{B_1}B_1^{-1}A_{.m+1}$ has a symmetric distribution. The probability of a symmetric continuous random variable being non-negative is 1/2. This can be easily seen from the following argument. Let Z be a continuous symmetric random variable, i.e., Z and $-Z$ have the same distribution. $P(Z \geq 0) = P(-Z \geq 0) = P(Z \leq 0) = 1 - P(Z \geq 0) = \frac{1}{2}$ Hence,

$$\begin{aligned} E(T) &= \binom{n}{m} E_{c_{B_1}B_1^{-1}} \frac{1}{2^{n-m}} \\ &= \binom{n}{m} \frac{1}{2^{n-m}}. \end{aligned}$$

Note that there is no loss of generality in assuming that the elements of coefficient matrix A and cost vector c, belong to the interval $[-1\ \ 1]$. However, in a typical problem, one does not expect to see the elements distributed uniformly. Moreover, real-life problems are more sparse and some of the coefficients have inherent interdependencies, so that the assumption of independence is not valid. Nevertheless it has been observed in real-life problems that the number of dual extreme points is remarkably small like randomly generated problems.

Given a particular sparse structure of a coefficient matrix and cost row of a particular problem, it is possible to get an idea of how big the dual number of extreme points is. The following corollary sheds some light on this aspect.

Corollary 1: Suppose the coefficient matrix A and cost row c have a given sparse structure. Assume that each non-zero element has a uniform distribution on the interval $[-1\ \ 1]$. Denote by S the number of non-singular submatrices of size $m \times m$

of A. Then, S is a constant with probability one and the expected number of dual extreme points equals $S/2^{n-m}$

Proof: Observe that the determinant of a given set of m columns from A has a determinant that equals the sum of products of different continuous random variables. (It could be an empty sum, making it possible that a given set of columns is singular). However, given that it is a not identically zero, the probability of it equalling zero is zero. So if a set of columns are non-singular for one realization of the random elements of A and c, then it is so with probability one. From the proof of the above theorem it follows that a given non-singular matrix corresponds to a dual extreme point with a probability of $\frac{1}{2^{n-m}}$. Therefore the expected number of extreme points for the dual problem is $S/2^{n-m}$.

Note that the above two proofs used only the fact that the distribution of the random elements in A and c are independent and symmetric around zero. In particular, they can be any other symmetric distribution, such as the Normal distribution, and each one can be a different symmetric distribution. Thus we have the following corollary.

Corollary 2: The above two results are true, when the elements of A and c are distributed independently (not necessarily identically) and symmetrically around zero.

Therefore, we have reason to believe that a function of the form described in (2.32) can be constructed in a computationally inexpensive way which will be a good approximation to the original function. Now, we'll look into the other three criteria that define a 'good' approximation function.

As we mentioned earlier regarding the additive approximation function, we need to be able to ensure that the piecewise-linear approximation function is non-negative. In practice, we know that the function $f(\omega)$ is non-negative. Even then, $\bar{\Gamma}$ in (2.32) need not be non-negative. However, if the dimension of the vector $\pi_2^{j_i}$ is bigger than the number of random parameters, r, then a vector $\pi_2^{j_0}$ can be found that makes $\pi_2^{j_0}(d(\omega) + B(\omega)x)$ positive for all ω. If we include this one in the approximation function, we will get a positive approximation function. This, in fact, can be done even if the function $f(\omega)$ takes on negative values. However, it is not advisable in that case. In such cases, it is better to split the function $\Gamma(\omega)$ into a positive part $\Gamma^+(\omega)$, and negative part $\Gamma^-(\omega)$ and rewrite θ as follows.

$$\begin{aligned}\theta &= \int_\Omega f(\omega)p(\omega)d\omega \\ &= \overline{\Gamma^+}\int_\Omega \frac{f(\omega)}{\Gamma(\omega)}\frac{\Gamma^+(\omega)p(\omega)}{\overline{\Gamma^+}}d\omega - \overline{\Gamma^-}\int_\Omega \frac{f(\omega)}{\Gamma(\omega)}\frac{\Gamma^-(\omega)p(\omega)}{\overline{\Gamma^-}}d\omega.\end{aligned} \tag{2.33}$$

This will work well if $\Gamma(\omega)$ is bounded away from zero with probability one. If that is not the case, we can simply add a positive constant to both $\Gamma^+(\omega)$ and to $\Gamma^-(\omega)$ to overcome the problem.

2.5.2. Ease of evaluation

It is clear that, being maximum of some linear functions, this function is extremely easy to evaluate. The original function we have is also a maximum of linear functions, i.e., a linear program. However, the fact that we don't know all of the linear pieces explicitly, makes it necessary to apply an algorithm like the Simplex algorithm to find it. That is more expensive than evaluation of our approximation function.

2.5.3. Evaluating the Importance Density

We want a sampling scheme that would choose an n-dimensional vector ω with probability $\frac{\Gamma(\omega)p(\omega)}{\bar{\Gamma}}$. When $\Gamma(\omega)$ is an additive function and $p(\omega)$ is the joint probability of independent discrete random variables, evaluating this is easy, as it involves evaluating n one-dimensional integrals to set-up new probabilities. But when the function Γ is the maximum of certain linear functions, as it is in our case, this is no longer easy. This is because evaluating $\bar{\Gamma}$ is a difficult problem in itself. Therefore we use the following three step sampling strategy to select ω with probability $\Gamma(\omega)p(\omega)/\bar{\Gamma}$, without evaluating $\bar{\Gamma}$ explicitly.

To do this, first find a function $u(\omega)$ that dominates the function $\Gamma(\omega)$. For example, $u(\omega) = M$ where M is an upper bound for $\Gamma(\omega)$. Let $\bar{u}$ be the expectation of $u(\omega)$ where ω has the density $p(\omega)$. Suppose for the moment, that we know how to select an ω with probability $\frac{u(\omega)p(\omega)}{\bar{u}}$. (This is used in Step 1 below and we'll discuss how to do this later).

- Step 1 Randomly choose $\omega = \omega_0$ with probability $u(\omega)p(\omega)/\bar{u}$

- Step 2 Randomly choose a t_0 uniformly from $[0\ u(\omega_0)]$.

- Step 3 If $\Gamma(\omega_0) < t_0$ then declare ω_0 as the sample point selected, otherwise reject ω_0 and go to step 1.

Let r be the probability of going through Steps 1 through 3 and coming back to Step 1 as one iteration. Let r denote the probability of one iteration. Also, let $q(\omega)$ be the probability of selecting an ω using the above three-step procedure.

$$
\begin{aligned}
q(\omega) &= \sum_{i=0}^{\infty} P(i \text{ rejections and accepting } \omega \text{ in } i+1\text{st cycle}) \\
&= \sum_{i=0}^{\infty} r^i \frac{p(\omega)u(\omega)}{\bar{u}} \frac{\Gamma(\omega)}{u(\omega)} \\
&= \frac{p(\omega)\Gamma(\omega)}{\bar{u}(1-r)}.
\end{aligned}
$$

From the above discussion it also follows that $r = 1 - \frac{\bar{\Gamma}}{\bar{u}}$ and $q(\omega) = \Gamma(\omega)p(\omega)/\bar{\Gamma}$ as required. From this we can see that the expected number of iterations the procedure takes is $\bar{u}/\bar{\Gamma}$. Therefore, the smaller the mean of the upperbound function, the less time consuming this method will be.

Now, coming back to Step 1, we have to specify, how to choose ω with probability $u(\omega)p(\omega)/\bar{u}$. This is exactly the same kind of problem we started with, i.e., to choose ω with probability $\Gamma(\omega)p(\omega)/\bar{\Gamma}$. Moreover, we are free to choose a function u that is easy to deal with. For example, we can take $u(\omega) = M$ where, M is an upper bound for the function $\Gamma(\omega)$. In this case, Step 1 requires choosing ω with probability $p(\omega)$. When $\Gamma(\omega) = \max_{1 \le i \le l} \pi_{i_i}\omega$ and $\pi_{i_i}\omega \ge 0$ with probability one, one can use $u(\omega) = \sum_{i=1}^{l} \pi_{i_i}\omega$ as an upperbound function. Since this is an additive function choosing an ω with probability $u(\omega)p(\omega)/\bar{u}$ is easy. It only involves computing n one-dimensional integrals. Therefore if $\bar{u}$ thus obtained is smaller than M, it is better to use this $u(\omega)$ in Step 1.

The following diagram shows using the maximum of $\Gamma(\omega)$ for sampling.

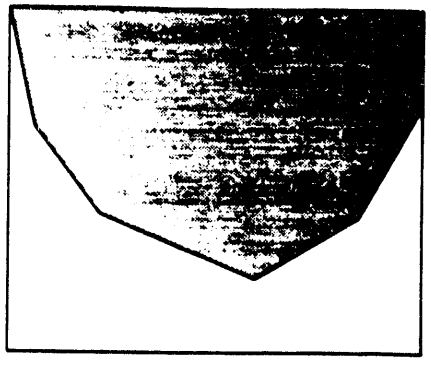


Fig 3

The shaded area is the area of rejection. It is clear that the rejection area is large for this particular case. In cases like this, one could divide the sample space into rectangles $R_1, \ldots, R_m$ and define

$$u(\omega) = \max_{y \in R_j} \Gamma(y) \text{ if } \omega \in R_j.$$

sampling with a function like this easy and it has a much smaller rejection region than the constant upper bound function. This is shown in the following diagram.

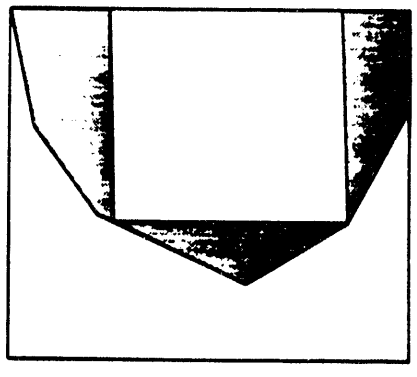


Fig 4

Therefore, we have shown that an explicit piecewise-linear function with small number of pieces is a good approximation function according to the first three criteria defined in (2.25) . Now, let us turn to the last criterion.

2.5.4. Evaluation of the mean of approximation

Recall from (2.8) that the mean of the approximation function plays a role in two different places. The first one is in determining the importance density $q(\omega)$. We have described how to achieve this without explicitly computing $\bar{\Gamma}$. The second role it plays is in the form of a multiplicative factor to $\bar{\rho}$ which is a sample mean for $\int_\Omega \rho(\omega)q(\omega)d\omega$. Hence, we do need to address this latter problem.

Because the approximation function that we propose is of the same structure as the original function, evaluating the mean of this function is clearly of the same nature as the problem we started out with. However, we know all the pieces of the approximate function which is not the case with the original function. Moreover, the approximate function has fewer pieces than the original function. It is these differences that we would like to take advantage of.

We consider two different approaches for evaluating the mean, $\bar{\Gamma}$ of the approximation function.

- Select Γ for which $\bar{\Gamma}$ can be easily evaluated exactly.

- Consider estimating $\bar{\Gamma}$ using sampling or other means.

Exact evaluation of $\bar{\Gamma}$:

Recall from (2.9) that $\theta = \bar{\Gamma}\bar{\rho}$. If we can evaluate $\bar{\Gamma}$ exactly, then an estimator for θ would be given by

$$\hat{\theta} = \bar{\Gamma}\hat{\rho} = \bar{\Gamma}\frac{1}{n}\left\{\rho(\omega_1) + \ldots \rho(\omega_n)\right\}$$

with an estimated variance of

$$\hat{\sigma_\theta^2} = \bar{\Gamma}^2 \frac{1}{n(n-1)} \sum_{i=1}^{n} (\rho(\omega) - \hat{\rho})^2$$

where $\omega_1, \ldots, \omega_n$ are chosen according to the importance density $q(\omega)$. Under certain conditions, it is possible to evaluate the mean of the approximation function exactly. For example, suppose $B(\omega)$ and $d(\omega)$ can be expressed linearly in the components of the vector ω as described in (2.12) and in (2.13) . This would enable us to express $\bar{\Gamma}$ as follows.

$$\begin{aligned}
\bar{\Gamma} &= \int_\Omega \Gamma(\omega) p(\omega) d\omega \\
&= \int_\Omega \max_{1 \le i \le l} \left\{ \pi_2^{j_i} \left(d_0 + B_0 x + \sum_{k=1}^{r_d} \omega^k d_k + \sum_{k=r_d+1}^{r} \omega^k (B_k x) \right) \right\} p(\omega) d\omega \\
&= \sum_{i=1}^{l} \int_{H_i} \pi_2^{j_i} \left\{ d_0 + B_0 x + \sum_{k=1}^{r_d} \omega^k d_k + \sum_{k=r_d+1}^{r} \omega^k (B_k x) \right\} p(\omega) d\omega \\
&\quad \text{where } H_i = \left\{ \omega \mid \pi_2^{j_i} \left(d(\omega) + B(\omega) x \right) \ge \pi_2^{j_s} \left(d(\omega) + B(\omega) x \right), \quad \forall \quad 1 \le s \le l \right\} \\
&= \sum_{i=1}^{l} \sum_{k=0}^{r} a_{ik} \int_{H_i} \omega^k p(\omega) d\omega \\
&\quad \text{where} \quad a_{ik} = \begin{cases} \pi_2^{j_i} (d_0 + B_0 x) & \text{if } k = 0 \\ \pi_2^{j_i} d_k & \text{if } 1 \le k \le r_d \\ \pi_2^{j_i} B_k x & \text{if } r_d < k \le r \end{cases} \qquad (2.34) \\
&\quad \omega^0 = 1.0 \\
&\quad \text{and } H_i = \left\{ \omega \mid \sum_{k=0}^{r} a_{ik} \omega^k \ge \sum_{k=0}^{r} a_{sk} \omega^k \quad \forall \quad 1 \le s \le r \right\}. \qquad (2.35)
\end{aligned}$$

Therefore, the problem of evaluating $\bar{\Gamma}$ has been reduced to that of integrating a linear function on a polyhedron. Suppose that the components of ω are independently distributed on a compact set with certain continuous densities. When this is the case, the intersection of the polyhedron H_i in (2.35) with the compact probability space gives us a polytope to work with. Thus we have the following problem.

$$\int_P \omega^i p(\omega) d\omega$$

$$\text{where } P = \left\{ \omega \,|\, \omega = \sum_{i=1}^{k} \lambda_i p_i;\ \sum_{i=1}^{k} \lambda_i = 1;\ \lambda \geq 0 \right\}$$

$$= \sum_{j=1}^{k-r} \int_{S_j} \omega^i p(\omega) d\omega$$

$$= \sum_{j=1}^{k-r} \int_{\{\omega \,|\, \omega = G_j \lambda;\ e^T \lambda \leq 1;\ \lambda \geq 0\}} \omega^i p(\omega) d\omega$$

$$= \sum_{j=1}^{k-r} \sum_{t=1}^{r} G_{it} |G_j| \int_{\{\lambda \,|\, e^T \lambda \leq 1;\ \lambda \geq 0\}} \lambda_t p(G_j \lambda) d\lambda. \tag{2.36}$$

The $S - j$'s in the above equations are simplicies that make up the polytop S, and G_j is the matrix corresponding to the simplex S_j. Hence, the original integral can be performed if the integral in (2.36) can be evaluated exactly. In the following discussion, we give examples of some densities $p(\omega)$ under which the integrals

$$\int_{\{\lambda \,|\, e^T \lambda \leq 1;\ \lambda \geq 0\}} \lambda_1 p(G\lambda) d\lambda$$

$$\int_{\{\lambda \,|\, e^T \lambda \leq 1;\ \lambda \geq 0\}} p(G\lambda) d\lambda$$

can be computed analytically, where G is a non-singular deterministic matrix.

Uniform Density

$$p(\lambda) = \text{Constant}$$

$$\int_{\{\lambda \mid e^T\lambda \leq 1, \lambda \geq 0\}} d\lambda = \frac{1}{r!}$$

$$\int_{\{\lambda \mid e^T\lambda \leq 1, \lambda \geq 0\}} \lambda_1 d\lambda = \frac{1}{(r+1)!} \tag{2.37}$$

Exponential Density

$$p(\lambda) = c * e^{-\sum_{i=1}^{r} b_i \lambda_i} \tag{2.38}$$

Let $Gb = a$.

$$\int_{\{\lambda \mid e^T\lambda \leq 1, \lambda \geq 0\}} e^{-\sum_{i=1}^{r} a_i \lambda_i} d\lambda = \frac{1}{\prod_{j=1}^{r} a_i} - \sum_{k=1}^{r} \frac{e^{-a_k}}{a_k \prod_{\substack{j=1 \\ j \neq k}}^{r} (a_j - a_k)}$$

$$\int_{\{\lambda \mid e^T\lambda \leq 1, \lambda \geq 0\}} \lambda_r e^{-\sum_{i=1}^{r} a_i \lambda_i} d\lambda = \frac{1}{a_r \prod_{j=1}^{r} a_i}$$

$$- \sum_{k=1}^{r-1} \frac{e^{-a_k}}{(a_k)(a_r - a_k) \prod_{\substack{j=1 \\ j \neq k}}^{r} (a_j - a_k)} \tag{2.39}$$

$$-e^{-a_r} \frac{1 + a_r}{a_r \prod_{j=1}^{r} a_i} + e^{-a_r} \sum_{k=1}^{r-1} \frac{1 + a_r - a_k}{(a_k)(a_r - a_k) \prod_{\substack{j=1 \\ j \neq k}}^{r-1} (a_j - a_k)}$$

Polynomial density

$$p(\lambda) = c * \prod_{j=1}^{n} \lambda_i^{k_j} \tag{2.40}$$

When the density is of this form, $p(G_j\lambda)$ will be in the form of a sum of certain terms, each of which is again in the form described in (2.40) . Therefore, it is enough to be able to integrate the function in (2.40) .

$$\int_{\{\lambda \mid e^T\lambda \leq 1, \lambda \geq 0\}} \prod_{j=1}^{n} \lambda_i^{k_j} d\lambda = \frac{k_1! + \ldots + k_n!}{(k_1 + \ldots + k_n + n)!}. \tag{2.41}$$

When the given exogenous probability density $p(\omega)$ does not take any of the above forms, it might still be possible to approximate the given probability density with one of the above. Thus, by expressing θ as,

$$\begin{aligned}\theta &= \int_\Omega c(\omega)p(\omega)d\omega \\ &= \left\{\int_\Omega \Gamma(\omega)g(\omega)d\omega\right\}\left\{\int_\Omega \frac{c(\omega)}{\Gamma(\omega)}\frac{p(\omega)}{g(\omega)}\frac{g(\omega)\Gamma(\omega)}{\bar{\Gamma}}d\omega\right\}\end{aligned}$$

we can approximate the function $f(\omega)$ by $\Gamma(\omega)$ and the probability $p(\omega)$ with $g(\omega)$ and successfully implement the importance sampling scheme. The biggest obstacle in this approach is the amount of work involved in breaking a polytope into a series of simplicies. In practice, this is a complex task and essentially forces us to abandon this approach in all but few cases. Therefore, we will now describe, how we can carry out importance sampling scheme without computing the exact value of $\bar{\Gamma}$.

Estimating the mean of the approximation function

Since we can not easily evaluate $\bar{\Gamma}$ exactly, we have to resort to approximation techniques. We can take one of the following two approaches.

- Find deterministic bounds for $\bar{\Gamma}$.

- Estimate $\bar{\Gamma}$ and obtain an estimate of its variance and construct a confidence interval.

If we get deterministic lower and upper bounds $\bar{\Gamma}_l$ and $\bar{\Gamma}_u$ for $\bar{\Gamma}$, and estimate $\bar{\rho}$ by $\hat{\rho}$ with an estimated variance of $\hat{\sigma}^2_{\hat{\rho}}$ then $\left[\bar{\Gamma}_l(\bar{\rho} - z_{\frac{p}{2}}\sigma_{\hat{\rho}})\ \bar{\Gamma}_u(\bar{\rho} + z_{\frac{p}{2}}\sigma_{\hat{\rho}})\right]$ is a $p\%$ confidence interval for θ. The sharper the bounds for $\bar{\Gamma}$ are, the smaller this interval will be.

If we estimate $\bar{\Gamma}$ by $\hat{\Gamma}$, with an estimated variance $\hat{\sigma}_{\hat{\Gamma}}$ and independently estimate $\bar{\rho}$ by $\hat{\rho}$ with an estimated variance $\sigma^2_{\hat{\rho}}$ then $\hat{\Gamma}\hat{\rho}$ is an unbiased estimator for θ with an estimated variance $\hat{\rho}^2\sigma^2_{\hat{\Gamma}} + \hat{\Gamma}^2\sigma^2_{\hat{\rho}} + \hat{\sigma}^2_{\hat{\Gamma}}\hat{\sigma}^2_{\hat{\rho}}$. Therefore, the square of coefficient of variation is given by

$$\begin{aligned} c^2_{\hat{\theta}} &= \frac{\hat{\sigma}^2_{\hat{\theta}}}{\hat{\theta}^2} \\ &= \frac{\hat{\rho}^2\sigma^2_{\hat{\Gamma}} + \hat{\Gamma}^2\sigma^2_{\hat{\rho}} + \hat{\sigma}^2_{\hat{\Gamma}}\hat{\sigma}^2_{\hat{\rho}}}{\hat{\Gamma}^2\hat{\rho}^2} \\ &= c^2_{\hat{\rho}} + c^2_{\hat{\Gamma}} + c^2_{\hat{\rho}}c^2_{\hat{\Gamma}}. \end{aligned}$$

Therefore, by designing an approximation function that approximates the real function very well, we can control $c^2_{\hat{\rho}}$ significantly. Then, the dominating factor in the variability of $\hat{\theta}$ would be the variability in $\hat{\Gamma}$. Therefore, we should find means of estimating $\bar{\Gamma}$ with as little variance as possible. This we attempted in three different ways.

- Apply other variance-reduction techniques.

- Apply importance sampling with additive approximation.

- Apply naive sampling with a very large sample size.

We have also experimented with the use of control-variate techniques as described below.

Let us rewrite $\bar{\Gamma}$ as follows.

$$\begin{aligned}\bar{\Gamma} &= \int_{\Omega} \Gamma(\omega)p(\omega)d\omega \\ &= \int_{\Omega} (\Gamma(\omega) - G(\omega))p(\omega)d\omega + \int_{\Omega} G(\omega)p(\omega)d\omega\end{aligned} \tag{2.42}$$

where $G(\omega)$ is an approximation for $\Gamma(\omega)$ and is positively correlated with it. We can also use a $G(\omega)$ whose expectation $\int_{\Omega} G(\omega)p(\omega)d\omega$ can be computed exactly. Thus $\bar{\Gamma}$ is estimated by

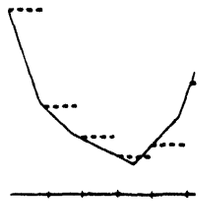
$$\frac{1}{n}\{\sum_{i=1}^{n}(\Gamma(\omega_1) - G(\omega_1)) + \ldots + (\Gamma(\omega_n) - G(\omega_n))\} + \bar{G} \tag{2.43}$$

where $\omega_1, \ldots, \omega_n$ are chosen according to the probability density $p(\omega)$. The variance of such an estimator is given by

$$\frac{1}{n}(\sigma_{\Gamma}^2 + \sigma_G^2 - 2\mathrm{COV}(\Gamma, G)).$$

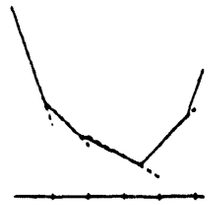
By a good choice of the function $G(\omega)$, this can be made much smaller than $\frac{1}{n}\sigma_{\Gamma}^2$. To choose a function than can be easily integrated, we observe that integrating a linear or constant function on hyper-rectangles is easy. Therefore, we divide the sample space Ω into a finitely many rectangles define $G(\omega)$ to be either constant or a linear function on each of the rectangles. Both these versions are described below and are shown in graphs.

$$g(\omega) = \sum_{i=1}^{nr} g_i I_{R_i}(\omega)$$



$$g(\omega) = \sum_{i=1}^{nr} \sum_{j=0}^{r} \omega^j a_{r_i j} I_{R_i}(\omega)$$

where $r_1, \ldots, r_{nr}$ is a sequence of integers between 1 and l, and $a_{r_i j}$ are as defined in (2.34) .



As our numerical results show, this is a good approach when the number of random parameters r is not too high. However, when it is too high , it becomes computationally too expensive to partition the sample space into a sufficiently fine partition to give a good $G(\omega)$.

When no special techniques can be devised to estimate $\bar{\Gamma}$, we resort to two of the methods we discussed earlier, i.e., importance sampling with additive approximation and naive sampling. It might look counterintuitive that an approach that did not work well with the estimation of θ will work with estimation of $\bar{\Gamma}$. However, one should note that, both naive sampling and usage of additive approximation will always give good results if sufficiently large sample sizes are easy to process. Using a higher sample size is a very expensive proposition when it comes to estimating θ, but not so in estimating $\bar{\Gamma}$.

2.6. Numerical Results

We have used the software developed by Dr. Gerd Infanger to solve two-stage stochastic linear programs. This software combines Benders decomposition with importance sampling, using an additive-approximation scheme. It also solves the universe problem (if possible), the expected value problem and uses an advanced importance sampling techniques like the exponential smoothing. It is written in FORTRAN and uses MINOS (Murtagh and Saunders (1982) [29]), as a subroutine to solve linear programs. We have modified the above software to incorporate the different implementations of importance sampling described in previous sections.

The extreme points $\pi_2^{j_i}$ in (2.32) are obtained by solving the linear program (2.10) for a predetermined number of scenarios ω and storing the corresponding dual extreme points. To select these scenarios we start out by using the scenario that assumes the smallest values of each of the random components. We then generate successive scenarios by replacing i-th component of the vector by the maximum of that random parameter, where i is increased from 1 to r. We also include the scenario consisting of the mean values of all the random parameters. In this way, we include the scenario that has all the smallest values, the scenario that has all the largest values and the scenario that has the mean values. If the predetermined number of scenarios is larger than $r + 2$, the rest of the scenarios are randomly generated according to the given probability distribution. A new $\pi_2^{j_i}$ is added to the list only if it is significantly different from the existing ones in a least squares sense.

In the implementation of piecewise-linear approximation scheme there are two sample sizes. Sample size n refers to the size used to estimate $\bar{\rho}$ and sample size

m refers to the size used to estimate $\bar{\Gamma}$. However, in the additive approximation scheme there is only one relevant sample size. Thus, plotting sample sizes against variances is not an appropriate comparison. Therefore, to compare these approximation schemes we compute the CPU time spent by the sampling routine of the program and record the length of the confidence interval produced. We used sample sizes, ranging from 20 to 500 for the additive approximation scheme and for the estimation of $\bar{\rho}$ in the piecewise-linear approximation scheme. Sample sizes, ranging from 500 to 50000 are used to estimate $\bar{\Gamma}$ in the the piecewise-linear approximation scheme.

We show numerical results from four problems that we solved. Each of them is formulated as a two-stage stochastic linear program. In each of the problems we have solved the expected value problem and estimated the expected second-stage costs the solution vector obtained. Results from five different strategies are compared. These are,

- Naive sampling
- Importance sampling with additive approximation
- Importance sampling with piecewise-linear approximation where $\bar{\Gamma}$ is estimated using naive sampling.
- Importance sampling with piecewise-linear approximation where $\bar{\Gamma}$ is estimated using a control-variate technique.
- Importance sampling with piecewise-linear approximation where $\bar{\Gamma}$ is estimated using an importance sampling approach using an additive approximation.

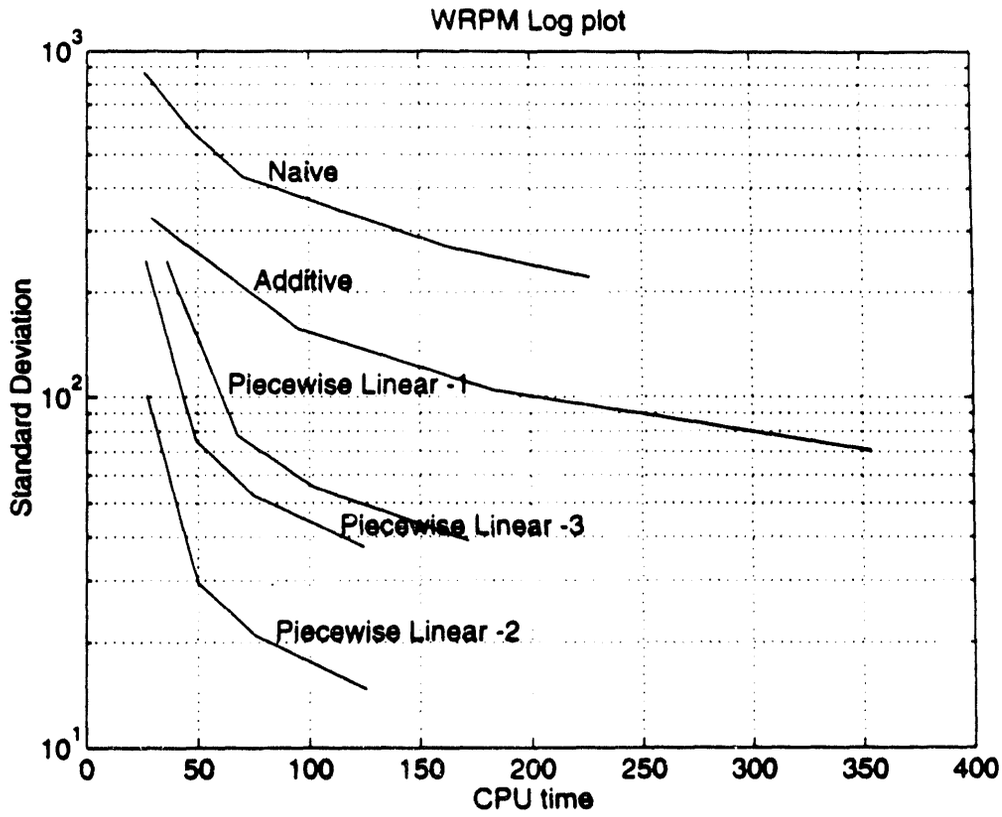
For each of the strategies, we have solved the problem with different sample sizes and computed the lengths of 95% confidence intervals and the CPU time taken for this. The CPU time does not involve the setup time for the problem, but includes the time to estimate the sample cut as described in previous chapter. Each of the problems is described in detail below and the results are shown in graphical and tabular form at the end.

Problem: WRPM

This is a multi-area capacity expansion planning problem for the western US and Canada. A detailed description of this model can be found in Dantzig et.al. (1989) [10]. The objective of the model is to determine optimum discounted least-cost levels of generation and transmission facilities for each region covered in the model over time. We considered a one period version of this model that fits into the two-stage stochastic linear programming framework. The random elements of the model are the availabilities of generators and transmission lines and the demands. The size of the subproblem and the number of scenarios are given in Table 9. The results from this problem are presented in Table 1 and Graph 4 .

Sample Size	$\hat{\theta}$	$\hat{\sigma}_{\hat{\theta}}$	CPU
Naive sampling			
100	0.1377927E+06	0.8682487E+03	0.2558000E+02
200	0.1379870E+06	0.5716234E+03	0.4878000E+02
300	0.1380090E+06	0.4293819E+03	0.7026000E+02
500	0.1381891E+06	0.3378813E+03	0.1173900E+03
700	0.1381084E+06	0.2711093E+03	0.1613600E+03
1000	0.1380742E+06	0.2199263E+03	0.2263900E+03
Additive approximation			
100	0.1382268E+06	0.3260952E+03	0.2902000E+02
400	0.1381554E+06	0.1570986E+03	0.9517000E+02
800	0.1380747E+06	0.1044069E+03	0.1835800E+03
1600	0.1380473E+06	0.7064273E+02	0.3533800E+03
Piecewise-linear approximation with naive sampling			
$n_s = 50, m_s = 1000$	0.1381816E+06	0.2456612E+03	0.2641000E+02
$n_s = 50, m_s = 10000$	0.1381055E+06	0.7501227E+02	0.4910000E+02
$n_s = 50, m_s = 20000$	0.1381026E+06	0.5260600E+02	0.7462000E+02
$n_s = 50, m_s = 40000$	0.1381041E+06	0.3741754E+02	0.1244800E+03
Piecewise-linear approximation with Importance sampling			
$n_s = 50, m_s = 1000$	0.1380821E+06	0.1009098E+03	0.2708000E+02
$n_s = 50, m_s = 10000$	0.1380743E+06	0.2944668E+02	0.5004000E+02
$n_s = 50, m_s = 20000$	0.1380749E+06	0.2090695E+02	0.7561000E+02
$n_s = 50, m_s = 40000$	0.1380752E+06	0.1468641E+02	0.1255700E+03
Piecewise-linear approximation with control variates			
$n_s = 50, m_s = 1000$	0.1379015E+06	0.2455194E+03	0.3584000E+02
$n_s = 50, m_s = 10000$	0.1380581E+06	0.7790319E+02	0.6753000E+02
$n_s = 50, m_s = 20000$	0.1380780E+06	0.5537660E+02	0.1019100E+03
$n_s = 50, m_s = 40000$	0.1380484E+06	0.3902989E+02	0.1716800E+03

Table 1


WRPM Log plot
Standard Deviation
10^3
10^2
10^1
0
50
100
150
200
250
300
350
400
CPU time
Naive
Additive
Piecewise Linear -1
Piecewise Linear -3
Piecewise Linear -2

Graph 1

Problem: CAP

A manufacturing plant produces m different part types on any one of n machines. Each machine j has a fixed number of hours , h_j, currently available per week. The number of hours of new capacity j acquired is denoted by x_j and costs c_j dollars per hour. The total usage of machine j should not exceed U_j hours per week. Each machine j requires t_j hours of maintenance for every hour of operation. The total scheduled maintenance for all machines is not to exceed T hours per week. Part i is produced on machine j at a rate of a_{ij} parts per hour at a cost of g_{ij}. Each week an order of w_i units of part i must be met. If the total demand exceeds the total capacity then the excess parts are obtained from a subcontractor at a premium price, p_i, for part i. It is assumed that $p_i \gg g_{ij}$ for all parts i and machines j. The weekly demands are treated as i.i.d. random variables with known distribution. With the objective of minimizing the cost of new capacity plus the expected operating costs plus the expected cost of meeting excess demand from the subcontractor, the above problem is formulated and solved as two-stage stochastic linear program.

Problem data

The above problem is solved with the following data.

$$\begin{aligned} m &= 3 \\ n &= 4 \\ c_j &= (2.5, 3.75, 5.0, 3.0) \\ t_j &= (0.08, 0.04, 0.02, 0.01) \\ T &= 100 \\ U_j &= (2000, 2000, 2000, 2000) \\ p_i &= (400, 400, 400) \end{aligned} \tag{2.44}$$

g_{ij}

$i \downarrow$ $j \rightarrow$	1	2	3	4
1	2.6	3.4	3.4	2.5
2	1.5	2.3	2.0	3.6
3	4.0	3.8	3.5	3.2

Table 2

a_{ij}

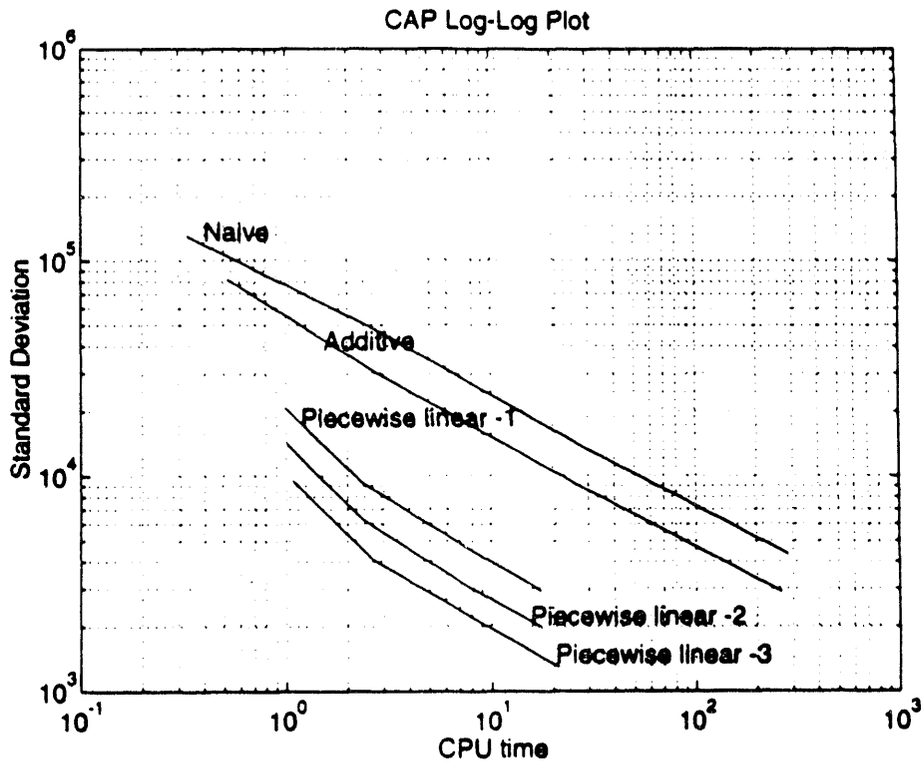
$i \downarrow$ $j \rightarrow$	1	2	3	4
1	0.6	0.6	0.9	0.8
2	0.1	0.9	0.6	0.8
3	0.05	0.2	0.5	0.8

Table 3

Each element of the random vector ω takes the values $300, 600, \ldots, 3000$ with probability of 0.1, independently of other elements. The results from this problem are presented in Table 4 and Graph 2.

Sample Size	$\hat{\theta}$	$\hat{\sigma}_{\hat{\theta}}$	CPU
Naive sampling			
20	0.2116483E+06	0.1307643E+06	0.3300000E+00
200	0.3028722E+06	0.4785451E+05	0.2730000E+01
2000	0.2608951E+06	0.1343476E+05	0.2901000E+02
20000	0.2603107E+06	0.4323950E+04	0.2850900E+03
Additive approximation			
20	0.2327514E+06	0.8196999E+05	0.5200000E+00
200	0.2898255E+06	0.3031747E+05	0.2740000E+01
2000	0.2618426E+06	0.9065203E+04	0.2663000E+02
20000	0.2614249E+06	0.2900156E+04	0.2623000E+03
Piecewise-linear approximation with naive sampling			
$n_s = 20, m_s = 1000$	0.2556709E+06	0.2082914E+05	0.9900000E+00
$n_s = 20, m_s = 5000$	0.2596231E+06	0.9348429E+04	0.2350000E+01
$n_s = 20, m_s = 20000$	0.2611956E+06	0.4647089E+04	0.7530000E+01
$n_s = 20, m_s = 50000$	0.2598432E+06	0.2928713E+04	0.1744000E+02
Piecewise linear approximation with Importance sampling			
$n_s = 20, m_s = 1000$	0.2566923E+06	0.1411260E+05	0.1010000E+01
$n_s = 20, m_s = 5000$	0.2579674E+06	0.6289137E+04	0.2370000E+01
$n_s = 20, m_s = 20000$	0.2605898E+06	0.3116273E+04	0.7410000E+01
$n_s = 20, m_s = 50000$	0.2600434E+06	0.1974844E+04	0.1761000E+02
Piecewise linear approximation with control variates			
$n_s = 20, m_s = 1000$	0.2655022E+06	0.9526401E+04	0.1080000E+01
$n_s = 20, m_s = 5000$	0.2599874E+06	0.4077356E+04	0.2660000E+01
$n_s = 20, m_s = 20000$	0.2603536E+06	0.2039508E+04	0.9160001E+01
$n_s = 20, m_s = 50000$	0.2596919E+06	0.1291688E+04	0.2121000E+02

Table 4


CAP Log-Log Plot
Standard Deviation
CPU time
Naive
Additive
Piecewise linear -1
Piecewise linear -2
Piecewise linear -3
10^6
10^5
10^4
10^3
10^{-1}
10^0
10^1
10^2
10^3

Graph 2

Problem: CAPGEN

This problem is a modified version of the previous problem. It has higher dimensions and higher number of scenarios. This modification is done to observe the effect of increased size of the problem, while the structure of the problem is kept the same. The description of the problem remains the same. The data is given below.

$$
\begin{aligned}
m &= 30 \\
n &= 10 \\
c_j &= (2.5, 3.75, 5.0, 3.0, 5.0, 2.5, 3.75, 5.0, 3.0, 5.0) \\
t_j &= (0.08, 0.04, 0.02, 0.01, 0.02, 0.08, 0.04, 0.02, 0.01, 0.02) \\
T &= 200 \\
h_j &= 500 \ \forall \ 1 \leq j \leq n \\
U_j &= (2000, 2000, 3000, 3000, 1000, 2000, 2000, 3000, 3000, 1000) \\
p_i &= 400 \ \forall \ 1 \leq i \leq m
\end{aligned}
\tag{2.45}
$$

g_{ij}

$i \downarrow \quad j \rightarrow$	1,5,9	2,6,10	3,7	4,8
$1, 4, \ldots, 28$	2.6	3.4	3.4	2.5
$2, 5, \ldots, 29$	1.5	2.3	2.0	3.6
$3, 6, \ldots, 30$	4.0	3.8	3.5	3.2

Table 5

a_{ij}

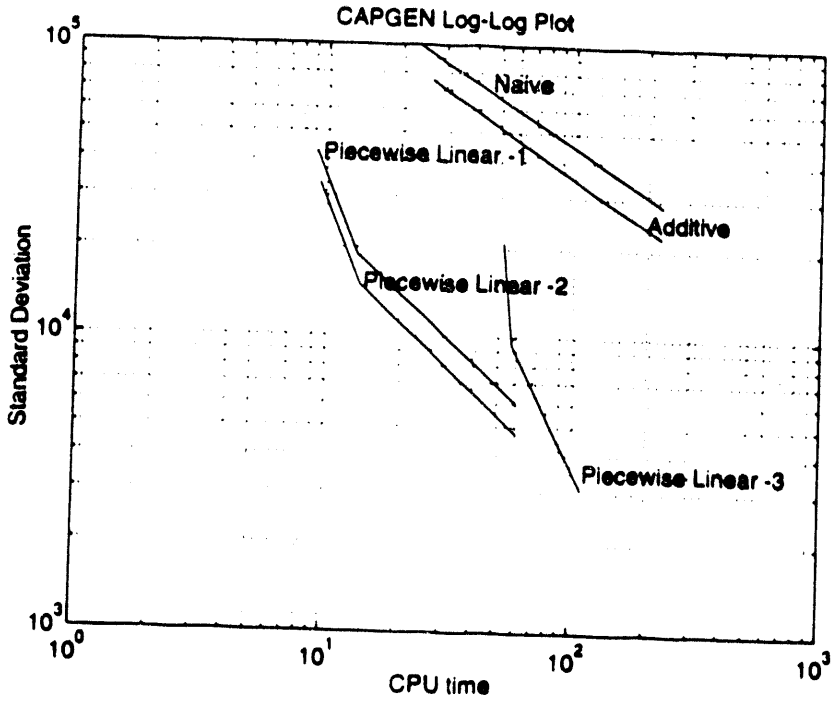
$i \downarrow \quad j \rightarrow$	1,5	2,6	3,7	4,8	9	10
$1, 4, \ldots, 28$	0.6	0.6	0.9	0.8	0.3	0.3
$2, 5, \ldots, 29$	0.1	0.9	0.6	0.8	0.8	0.5
$3, 6, \ldots, 30$	0.05	0.2	0.5	0.8	0.2	0.9

Table 6

The results from this problem are presented in Table 7 and Graph 3.

Sample Size	$\hat{\theta}$	$\hat{\sigma}_{\hat{\theta}}$	CPU
Naive sampling			
200	0.7227804E+06	0.9912831E+05	0.2335000E+02
1000	0.6759398E+06	0.4146293E+05	0.1151100E+03
2000	0.6544522E+06	0.2848225E+05	0.2285600E+03
Additive approximation			
200	0.6999338E+06	0.7569911E+05	0.2606000E+02
1000	0.6706608E+06	0.3181763E+05	0.1158500E+03
2000	0.6499043E+06	0.2231446E+05	0.2260300E+03
Piecewise linear approximation with naive sampling			
$n_s = 20, m_s = 1000$	0.6483633E+06	0.4261268E+05	0.9110001E+01
$n_s = 20, m_s = 5000$	0.6435018E+06	0.1904751E+05	0.1339000E+02
$n_s = 20, m_s = 50000$	0.6448479E+06	0.5983151E+04	0.6035000E+02
Piecewise linear approximation with Importance sampling			
$n_s = 20, m_s = 1000$	0.6470257E+06	0.3322023E+05	0.9420000E+01
$n_s = 20, m_s = 5000$	0.6413708E+06	0.1496077E+05	0.1381000E+02
$n_s = 20, m_s = 50000$	0.6439548E+06	0.4718087E+04	0.6050000E+02
Piecewise linear approximation with control variates			
$n_s = 20, m_s = 1000$	0.6278739E+06	0.2099258E+05	0.5221000E+02
$n_s = 20, m_s = 5000$	0.6395258E+06	0.9475039E+04	0.5728000E+02
$n_s = 20, m_s = 50000$	0.6422440E+06	0.3062884E+04	0.1102500E+03

Table 7


CAPGEN Log-Log Plot
Naive
Piecewise Linear -1
Additive
Piecewise Linear -2
Piecewise Linear -3
Standard Deviation
CPU time
10^5
10^4
10^3
10^0
10^1
10^2
10^3

Graph 3

Problem: RAND This is a randomly generated test problem. We have built a problem generator that generates problems of the form

$$
\begin{array}{llllllll}
\min & c_m x_m + c_{m+} x_{m+} + c_{m-} x_{m-} + E(f_s y_s(\omega) + f_{s+} y_{s+}(\omega) + f_{s-} y_{s-}(\omega)) & & & & & & \\
\text{s.t.} & Ax_m + & x_{m+} - & x_{m-} & & & = & b_1 \\
 & x_m & & & & & \leq & U_m \\
 & -Bx_m + & & & Dy_s(\omega) + & y_{s+}(\omega) - & y_{s-}(\omega) = & d \\
 & & & & y_s(\omega) & & \leq & U_s \\
 & & & \text{all variables are non-negative.} & & & &
\end{array}
\tag{2.46}
$$

It takes as input, the dimensions of the matrix A, B, D, the sparsity of the matrices A, B, D, the penalty cost vectors $c_{m+}, c_{m-}, c_{s+}, c_{s-}$ and the upperbounds U_m, U_s. The rest of the cost vector and the right hand side are generated as dense vectors, with the individual elements picked randomly from the interval $(0, 1)$. Every element of the matrices A, B, D is given a probability, equal to the specified sparsity, of being non-zero. These non-zero elements are randomly picked from the interval $(-1, 1)$.

In this specific problem, we have the following data.

dimensions of $A = 50 \times 75$

sparsity of $A = 0.15$

dimensions of $B = 50 \times 75$

sparsity of $B = 0.05$

dimensions of $D = 50 \times 75$

$$\text{sparsity of } D = 0.15$$

$$c_{m+}, c_{m-}, c_{s+}, c_{s-} = [100, \ldots, 100]$$

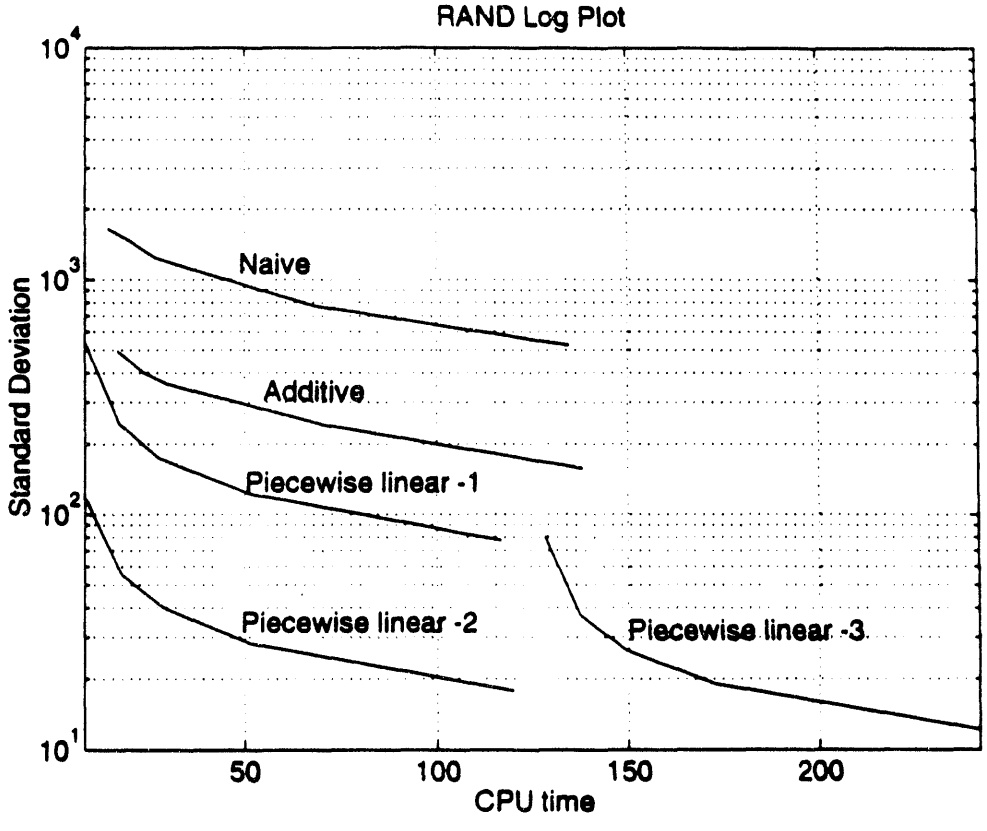
$$U_m = [\infty, \ldots, \infty]$$

$$U_s = [5, \ldots, 5].$$

The results from this problem are presented in Table 8 and Graph 4.

Sample Size	$\hat{\theta}$	$\hat{\sigma}_{\hat{\theta}}$	CPU
Naive sampling			
100	0.5214188E+05	0.1646190E+04	0.1424000E+02
150	0.5231466E+05	0.1446825E+04	0.2033000E+02
200	0.5236909E+05	0.1229894E+04	0.2723000E+02
500	0.5188471E+05	0.7712272E+03	0.6799000E+02
1000	0.5204281E+05	0.5244157E+03	0.1350400E+03
Additive approximation			
100	0.5179057E+05	0.4954130E+03	0.1672000E+02
150	0.5178824E+05	0.4033533E+03	0.2359000E+02
200	0.5178476E+05	0.3587222E+03	0.2994000E+02
500	0.5170700E+05	0.2396645E+03	0.7095000E+02
1000	0.5177767E+05	0.1571604E+03	0.1383300E+03
Piecewise-linear approximation with naive sampling			
$n_s = 20, m_s = 1000$	0.5190968E+05	0.5438035E+03	0.8389999E+01
$n_s = 20, m_s = 5000$	0.5177486E+05	0.2436146E+03	0.1724000E+02
$n_s = 20, m_s = 10000$	0.5160949E+05	0.1738789E+03	0.2794000E+02
$n_s = 20, m_s = 20000$	0.5171334E+05	0.1226858E+03	0.5120000E+02
$n_s = 20, m_s = 50000$	0.5175249E+05	0.7731601E+02	0.1167300E+03
Piecewise-linear approximation with Importance sampling			
$n_s = 20, m_s = 1000$	0.5177416E+05	0.1163012E+03	0.8710001E+01
$n_s = 20, m_s = 5000$	0.5173189E+05	0.5525127E+02	0.1789000E+02
$n_s = 20, m_s = 10000$	0.5168477E+05	0.3995796E+02	0.2926000E+02
$n_s = 20, m_s = 20000$	0.5171093E+05	0.2822614E+02	0.5132000E+02
$n_s = 20, m_s = 50000$	0.5172611E+05	0.1776708E+02	0.1198400E+03
Piecewise-linear approximation with control variates			
$n_s = 20, m_s = 1000$	0.5166554E+05	0.8041535E+02	0.1286500E+03
$n_s = 20, m_s = 5000$	0.5170490E+05	0.3714952E+02	0.1376700E+03
$n_s = 20, m_s = 10000$	0.5169956E+05	0.2637908E+02	0.1493100E+03
$n_s = 20, m_s = 20000$	0.5172025E+05	0.1896132E+02	0.1726100E+03
$n_s = 20, m_s = 50000$	0.5171697E+05	0.1218426E+02	0.2422100E+03

Table 8


RAND Log Plot
Standard Deviation
10^4
10^3
10^2
10^1
Naive
Additive
Piecewise linear -1
Piecewise linear -2
Piecewise linear -3
50
100
150
200
CPU time

Graph 4

These results are summarized in the Table 9. Let v_{sp} be the least variance obtained through strategy s on a specific problem p. Let t_{sp} be the time taken to achieve this variance. Let t_{nsp} be the amount of time that naive sampling would have taken to achieve a variance of v_{sp}. Then the number $\frac{t_{nsp}}{t_{sp}}$ represents how much more efficient strategy s is, for problem p. We have given these numbers in the following table, for the previous four problems and for four of the five strategies mentioned earlier. Observe that, using a piecewise-linear approximation function with additive approximation and with control variates seem to be superior to the other strategies. The efficiency of the control variate technique as we implemented, depends on the underlying grid size mentioned in section 5.4. Clearly, this becomes more and more expensive as the number of random parameters increases. This is reflected in the two problems CAP and CAPGEN.

In conclusion, we have explored different ways of implementing importance sampling strategies for stochastic programming problems. Different implementations perform with different levels of efficiency on different problems. However, we have seen that it is always better to reduce the integrand from a mathematical program to an explicit function.

SUMMARY

Problem	Size	Scenarios	Additive	Piecewise-2	Piecewise-3
WRPM	302 x 289	$3^5 4^6$	6.21	404.29	41.86
CAP	9 x 16	10^3	2.42	77.61	150.62
CAPGEN	30 x 300	10^{10}	1.54	137.68	179.27
RAND	50 x 75	5^{10}	10.87	981.70	1430.00

Piecewise-2 : Piecewise linear function used with additive approximation

Piecewise-3 : Piecewise linear function used with control variate method

Table 9

Chapter III
Decomposition methods

3.1. Introduction

Decomposition methods have been in use for a number of years, to solve large scale optimization problems. These have been motivated by problems of a special structure, such as problems with a stair case structure. Techniques that decompose and solve the problem , generally speaking, are far superior to techniques that do not. Sometimes, the size of the problem is so large that it is not possible to solve the original problem without decomposing it into smaller problems. For example, the deterministic equivalent of a typical two-stage stochastic linear program will have so many scenarios that they cannot be solved without some kind of a decomposition scheme.

Different kinds of decomposition techniques have been investigated in the literature. Notable among them are Dantzig-Wolfe decomposition or the Primal Decomposition scheme, (Dantzig (1960) [9]), Benders decomposition or the Dual Decomposition scheme, (Benders (1962) [4]), A Nested Decomposition approach for solving staircase structured linear programs due to Abrahamson (1983) [1], A Nested Decomposition algorithm for stochastic linear programs due to Birge (1985) [5], etc. The underlying principles in various decomposition schemes are outer linearization and inner linearization, as explained in, Geoffrion (1970) [19].

In this chapter, we introduce a Primal-Dual decomposition technique and compare it with the existing techniques. We plan to use it in the context of two-stage

stochastic linear programming. However, as we describe it we illustrate its use to solve the 'expected value problem' associated with the stochastic program.

This chapter is organized as follows. In section 2 we discuss Benders decomposition for linear programs. In section 3 we introduce a new approach called Primal-Dual decomposition. We then discuss some of its properties and describe why it is an improvement over Benders decomposition. Section 4 discusses various implementation issues and strategies. Finally in section 5, we present numerical results and directions for future research.

3.2. Benders Decomposition

Consider the two-stage stochastic linear program described in (1.2) .

The expected value problem is obtained by replacing all the random elements by their expectations. Thus we have the problem:

$$\begin{aligned} z^* = \min \quad & cx + c_2 y \\ \text{s.t.} \quad & Ax \qquad\quad = b \\ & -\bar{B}x + Dy = \bar{d} \\ & x, y \geq 0 \end{aligned} \tag{3.1}$$

where $\bar{B} = \sum_{\omega \in \Omega} p(\omega) B(\omega), \bar{d} = \sum_{\omega \in \Omega} p(\omega) d(\omega)$. In their current methodology to solve stochastic linear programs, Dantzig, et.al. [10], use Benders decomposition to solve the above problem, see also [35]. This method can be described briefly as follows.

Let us first define the recourse function $g(.)$ for the expected value problem and show that under reasonable assumptions, it is finite. We assume that the problem in (1.2) has full recourse, i.e., for every $\omega \in \Omega$ and x satisfying $Ax = b, x \geq 0$, there is a $y(\omega)$ such that $Dy(\omega) = d(\omega) + B(\omega)x, \qquad y(\omega) \geq 0$. In practice, this can be ensured by incorporating appropriate penalty variables into the recourse function. Clearly, this assumption implies that for every x satisfying $Ax = b, x \geq 0$ there is a y such that $Dy = \bar{d} + \bar{B}x, y \geq 0$. Therefore the 'recourse function for the expected value problem' defined by

$$\begin{aligned} g(x) = \min c_2 y \\ \text{s.t. } Dy = \bar{d} + \bar{B}x \\ y \geq 0 \end{aligned} \tag{3.2}$$

is less than $+\infty$ at every x satisfying $Ax = b, x \geq 0$. Similarly, we assume that the recourse function $Q(x, \omega)$ in (1.3) is bigger than $-\infty$ at some $\omega \in \Omega$ and x satisfying $Ax = b, x \geq 0$. This is equivalent to assuming that the dual of the linear program in (1.3) is feasible. Since that feasible region is the same as the feasible region of the dual in (3.2) , we know that $g(.)$ is bigger than $-\infty$. Therefore, the function $g(.)$ is finite at every x that satisfies $Ax = b, x \geq 0$.

Problem (3.1) can be shown to be equivalent to :

$$\begin{aligned} z^* = \min \quad & cx + \theta \\ \text{s.t.} \quad & Ax = b \\ & -g(x) + \theta \geq 0. \\ & x \geq 0 \end{aligned}$$

Let us use the dual representation of $g(.)$ given by,

$$\begin{aligned} g(x) = \max \pi(\bar{d} + \bar{B}x) \\ \text{s.t. } \pi D \leq c_2. \end{aligned}$$

Thus (3.1) is equivalent to

$$\begin{aligned} z^* = \min \quad & cx + \theta \\ \text{s.t.} \quad & Ax = b \\ & -\pi\bar{B}x + \theta \geq \pi\bar{d} \quad \forall \pi \text{ s.t. } \pi D \leq c_2 \\ & x \geq 0. \end{aligned} \tag{3.3}$$

Under Benders decomposition scheme, the set of constraints

$$-\pi\bar{B}x + \theta \geq \pi\bar{d} \quad \forall \pi \text{ s.t. } \pi D \leq c_2$$

are augmented by one per iteration. An iteration is defined as a pair of problems called master problem and Sub Problem which will be defined latter. These constraints are also called cuts, as they cut away part of the feasible region of the master problem before the augmentaion. Problem (3.3) with only $t(\geq 0)$ of the cuts generated, is called the master problem at iteration t. This problem has the following form after the t-th iteration for $t \geq 1$.

$$\begin{aligned} z_l^t = \min \quad & cx + \theta \\ \text{s.t.} \quad & Ax = b \\ & -\pi^{i^*}\bar{B}x + \theta \geq \pi^{i^*}\bar{d} \qquad \text{for } 1 \leq i \leq t \\ & x \geq 0 \end{aligned} \tag{3.4}$$

when $t > 0$. When $t = 0$, no cuts are placed and the variable θ is temporarily ignored. Let x^{t^*}, θ^{t^*} be the optimal solution vector of this problem.

This represents solving the original problem with the recourse function $g(.)$ replaced by an outer approximation given by

$$\bar{g}_l^t(x) = \max_{1 \leq i \leq t} \pi^{i^*}(\bar{d} + \bar{B}x).$$

The π^{t^*} to be used in the t-th cut, where $t \geq 1$, is generated from an optimization problem which is called the Sub Problem at iteration $t-1$. This problem has the following form.

$$\begin{aligned} g(x^{(t-1)^*}) = \min c_2 y & \\ \text{s.t. } Dy &= \bar{d} + \bar{B}x^{(t-1)^*} \\ y &\geq 0 \end{aligned}$$

where $x^{(t-1)^*}$ is the solution of the master problem at the $(t-1)$-th iteration. This subproblem at the $(t-1)$-th iteration represents the computation of the second stage cost of the first stage decision $x^{(t-1)^*}$. The optimal dual solution for this subproblem will be used in updating the outer approximation of the recourse function in the t-th iteration, and will be called π^{t^*}.

Thus one solves a sequence of master problems and subproblems alternatingly. The master and subproblems interact by sending the necessary information to each other at every iteration. We send the optimal decision variable, x^{t^*}, of the master problem of iteration t to the righthand side of the subproblem of the t-th iteration. The optimal dual vector for the subproblem of the t-th iteration, π^{t^*}, will be used in the $(t+1)$-st master problem as a new constraint given by:

$$-\pi^{t^*}\bar{B}x + \theta \geq \pi^{t^*}\bar{d}.$$

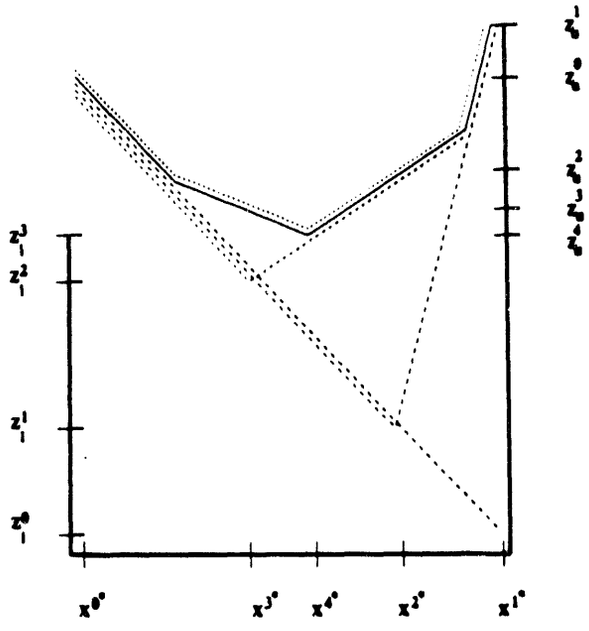
Moreover, we can generate lower and upper bounds for the optimum value z^* from the master and subproblems. As the master problem in (3.4) is a relaxation of the (3.3) , the optimum value z_l^t is a lower bound for z^*. Moreover, as the successive master problems are obtained by restricting the feasible region by imposing cuts, it is clear that the lower bounds form a non-decreasing sequence. Thus we have

$$z_l^0 \leq z_l^1 \ldots \leq z^*.$$

On the other hand every x^{t^*} is a first stage feasible decision and so

$$z_u^t = cx^{t^*} + g(x^{t^*})$$

is an upper bound for z^*. However, unlike the lower bounds the upper bounds need not form a monotone sequence. This is one of the ways in which the new scheme that we propose is going to be better. Another way it is better is demonstrated by the following picture.



Benders Decomposition

Fig 1

3.3. Primal-Dual Decomposition

To achieve faster convergence we modify the Benders decomposition scheme for linear problems as described below. In the standard method, at every iteration

t, we send the optimal solution of the master problem, x^{t^*}, to the subproblem to evaluate the second stage costs. In the modified version, we maintain the first stage decision vector at which the best upper bound has been found, say x_u^t, in the first t iterations. Instead of sending just one first stage decision to the subproblem, we send a variable linear convex combination of both the decision vectors, x^{t^*}, x_u^t. We modify the objective function of subproblem from evaluating the second stage costs at a given decision, to evaluating the best total of first and second stage costs at a convex combination of x^{t^*}, x_u^t. We define $x_u^{(t+1)}$ as this best convex combination of x^{t^*} and x_u^t. Thus, the subproblem at iteration t is of the following form:

$$\begin{aligned} z_u^t = \min \quad & (cx^{t^*})\lambda_t + (cx_u^t)\mu_t + c_2 y \\ \text{s.t.} \; & -(\bar{B}x^{t^*})\lambda_t - (\bar{B}x_u^t)\mu_t + Dy = \bar{d} \\ & \lambda_t + \qquad \mu_t \qquad = 1 \\ & \lambda_t \geq 0, \qquad \mu_t \geq 0, y \geq 0 \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} x_u^t &= \lambda^*_{(t-1)} x^{(t-1)^*} + \mu^*_{(t-1)} x_u^{(t-1)} \text{ for } t > 0 \\ &= x^{0^*} \text{ for } t = 0 \end{aligned}$$

where $x^{(t-1)^*}$ is the first stage decision vector obtained from the solution of $(t-1)$-st master problem and $\lambda^*_{(t-1)}, \mu^*_{(t-1)}$ are the weights obtained from the optimal solution of $(t-1)$-st Sub Problem.

The master problem still has the same structure as expressed in (3.4) . However, the π^{t^*} in (3.4) is now only part of the optimal dual vector of the subproblem. Let (π^{t^*}, η^{t^*}) be the complete set of dual multipliers for the subproblem given in (3.5) , where η^{t^*} corresponds to the last constraint and π^{t^*} corresponds to the rest of the constraints. These satisfy the dual constraints given by:

$$\begin{aligned} \pi^{t^*} D &\leq c_2 \\ \eta^{t^*} - \pi^{t^*} \bar{B} x_u^t &\leq c x_u^t \; . \\ \eta^{t^*} - \pi^{t^*} \bar{B} x^{t^*} &\leq c x^{t^*} \end{aligned} \tag{3.6}$$

As π^{t^*}'s generated by this method satisfy the constraints $\pi D \leq c_2$, it is valid to use them in the master problem. This scheme is depicted in Figure 2.

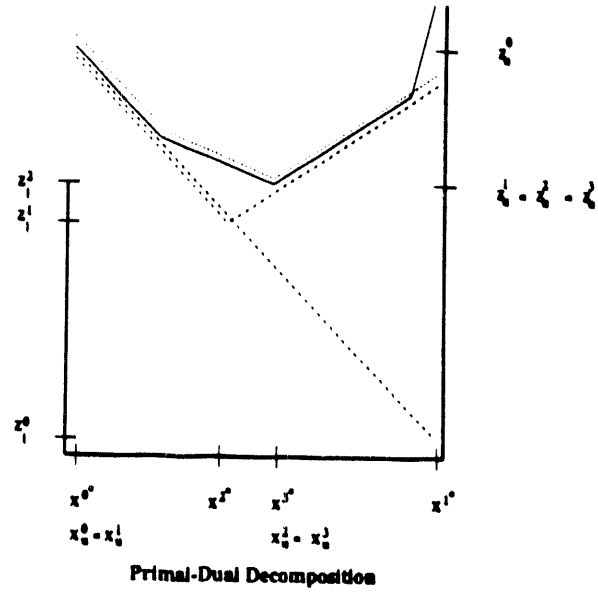


Fig 2

Note that, in Figure 2, we have the optimal solution at the end of iteration 1. in the form of x_u^2. This will always be true in the trivial cases when the first stage decision vector x has only one component, as is the case in the above graph. In the more realistic cases, we will not achieve optimality so quickly. We call this scheme the Primal-Dual decomposition scheme.

From this description of the scheme, we clearly have the following result.

Theorem 3.1: In Primal-Dual decomposition scheme, the upper bounds z_u^t are monotonically decreasing in t and the lower bounds z_l^t are monotonically increasing in t.

Since we have changed the decomposition scheme, we need to prove its convergence. Consider the problem described in Figure 2. At the end of iteration one

we got the optimal vector x_u^2, however we did not have enough cuts at that point to realize that it was an optimal solution. In other words, our upper bound had converged to z^* but the lower bound had not. Therefore we proceeded to place new cuts. The π^{t*} obtained by solving (3.5) is a subgradient of the function

$$cx + g(x)$$

at the point x_u^t. However, it might not be an extreme point of the set

$$\{\pi \mid \pi D \leq c_2\}.$$

Therefore, if we implemented the scheme exactly as we described, the cuts we would have obtained are likely to be as shown in Figure 3.

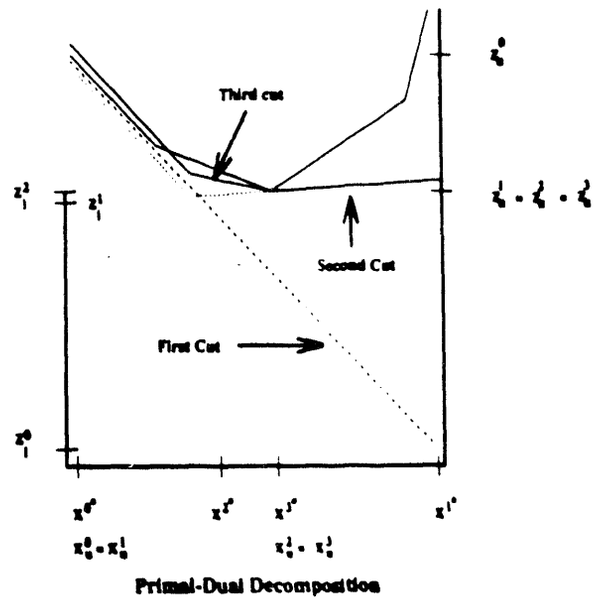


Fig 3

In other words, we could get cuts that do not coincide with any of the 'pieces' of the given piecewise linear function, such as the 2nd cut. Because of this, the

convergence of the lower bounds could be slow and it is even possible that the algorithm will not stop in finite time. However, by changing the implementation of this scheme, we can easily ensure not only that the algorithm converges, but it does so much faster than the Benders decomposition. These modifications are described in the next section. In the reminder of this section we describe a natural extension of the above scheme and justify the name 'Primal-Dual Decomposition'.

We could accelerate the convergence of the algorithm by passing x_u^t ,that gives the best upper bound in the first t iterations, to the $(t+1)$-st subproblem. This solution x_u^t is a convex combination of solutions of the first t master problems. Thus, it is natural to investigate the effects of passing $x^{0^*}, \ldots, x^{t^*}$, instead of just one convex combination of them, to the $(t+1)$-st subproblem. In other words consider the following subproblem

$$\begin{aligned} z_u^t = \min \quad & (cx^{1^*})\mu_1 + \ldots + (cx^{t^*})\mu_t + c_2 y \\ \text{s.t.} \quad & -(\bar{B}x^{1^*})\mu_1 - \ldots - (\bar{B}x^{t^*})\mu_t + Dy = \bar{d} \\ & \mu_1 + \ldots + \mu_t = 1 \\ & \mu_1, \ldots, \mu_t \geq 0, y \geq 0. \end{aligned} \tag{3.7}$$

Theorem 3.2: Let (P) be the primal problem described in (3.1) and (D) denote its dual. Let (S) be the subproblem described in (3.7) under the new decomposition scheme and (M) be the master problem described in (3.4) . If we use the same decomposition scheme on (D), the resulting master problem (M') and (S') are respectively the duals of (S) and (M).

Proof: The dual of (P) is given by

$$\begin{aligned} z^* = \min \quad & \pi\bar{d} + \eta b \\ \text{s.t.} \quad & \pi D \leq c_2 \\ & -\pi\bar{B} + \eta A \leq c. \end{aligned} \tag{3.8}$$

Notice that this has the same structure as the problem in (3.1) . Therefore, applying the new decomposition scheme, we get the following master and subproblems.

$$
\begin{aligned}
z_u^t = \min \quad & \pi\bar{d} + \delta \\
\text{s.t.} \quad & \pi D = c_2 \\
& -\pi\bar{B}x^{i^*} + \delta \leq cx^{i^*} \qquad \text{for } 1 \leq i \leq t
\end{aligned}
\tag{3.9}
$$

where x^{i^*} are dual multipliers corresponding to the first set of constraints in the following Sub problem.

$$
\begin{aligned}
z_l^t = \min \quad & (\pi^{1^*})\gamma_1 + \ldots + (\pi^{t^*})\gamma_t + \eta b \\
\text{s.t.} \quad & -(\pi^{1^*}\bar{B})\gamma_1 - \ldots - (\pi^{t^*}\bar{B})\gamma_t + \eta A \leq c \\
& \gamma_1 + \ldots + \gamma_t = 1 \\
& \gamma_1, \ldots, \gamma_t \geq 0, y \geq 0
\end{aligned}
\tag{3.10}
$$

where π^{i^*} is the solution of the i-th master problem described in (3.9) . Note that, since the problem in (3.8) is [illegible]ximization problem, the inequality sign in the cuts in the master problem is n[illegible]d we get upper bounds from the master problem and lower bounds from the subproblem. From the above it is clear that (3.10) is the dual of (3.4) and (3.9) is the dual of (3.7) . Hence the theorem is proved.

Note that in general, applying Benders decomposition on the primal is equivalent to applying Dantzig-Wolf decomposition on the dual. However, from the above theorem, we see that this new scheme remains same, whether applied to the primal or dual problems. Therefore we call this Primal-Dual decomposition.

3.4. Implementation Issues

We have implemented the scheme described in (3.7) . The following modifications are made to ensure convergence and to speed up the algorithm. If at iteration $(t+1)$, the lower bound has not increased by a significant amount, we try one of the three following methods to pick a 'good' π to be sent to the master problem.

In the first attempt, we try to pick a π that has a small absolute value, by obtaining a good solution for the following problem.

$$\begin{array}{ll} \min & \pi^T \pi \\ \text{s.t} & \pi D \leq c_2 \\ & \pi(\bar{d} + \sum_{i=0}^{t} \mu_i^* x^{i*}) = z_u^t. \end{array} \tag{3.11}$$

Observe that, this problem has all the subgradients of the function $cx + g(x)$ at $\sum_{i=0}^{t} \mu_i^* x^{i*}$ as the feasible region and tried to select a π that is as close to the null vector as possible. However, it is not necessary to solve this problem to optimality. We use an outer linearization and solve the corresponding linear program once. to get a π which is different from the previous one. It is also computationally convenient to do this since we need to make only a few changes to the subproblem to achieve this.

If incorporating the resulting π did not yield a better lower bound, we resolve the subproblem. This time, we only send $\sum_{i=0}^{t} \mu_i^* x^{i*}$ as input to the subproblem. This would ensure that the π that we get is an extreme point of the set $\{\pi \mid \pi D \leq c_2\}$. With this strategy, the only way this algorithm can stall, or even cycle, is if we choose the same extreme point over different iterations. If we notice this, i.e., if we have the same upper bound for 5 or more iterations, we resolve the subproblem by

sending a randomly perturbed solution $\sum_{i=0}^{t} \tilde{\mu}_i^* x^{i*}$ to the subproblem. This would have the effect of choosing one of the extreme points of $\{\pi \mid \pi D \leq c_2\}$ which is a subgradient at $\sum_{i=0}^{t} \mu_i^* x^{i*}$. Since there are only finitely many such extreme points, with probability one we will eventually pass all the necessary extreme points of $\{\pi \mid \pi D \leq c_2\}$ to the master problem.

While, these strategies have to be incorporated into the algorithm to ensure theoretical convergence, we do not often need them in practice. To measure the real work involved, we count every extra sub problem solved as an extra iteration.

In practice, we do not incorporate solutions of all the previous master problems into the current subproblem. That could blow up the size of the subproblem and render the usage of Decomposition moot. Therefore, we only make room for a pre-specified number, usually about 10, of master problem solutions to be incorporated into the current subproblem. This necessitates a scheme to choose which solutions to discard if we have more than we have space for. We use the following hueristics to determine that.

(1) When discarding a former solution, we discard the one with the smallest weight in the previous solution of the subproblem. In case of a tie, we discard the one among them that entered the Sub problem first.

(2) If we discard a solution that has positive weight in the previous iteration, we retain the best linear combination of solutions that gave the current upper bound. This enables us to maintain monotonicity of upper bounds.

We compared Primal-Dual decomposition scheme with Benders decomposition. The results from three different problems are presented in the following table. First problem WRPM is a capacity expansion planning problem, described in the previous

chapter. This has 2 rows and 34 columns in the master problem and 302 rows and 289 columns in the subproblem. HYDRO-1 is a two-stage hydroelectric scheduling problem in which the subproblem has 31 rows, and 158 columns in the first iteration and the master problem has 31 rows and 158 columns in the first iteration. A detailed description of this problem is available in Morton (1993) [28]. The last problem SSN is in the area of telecommunications capacity planning in which the subproblem has 176 rows, and 706 columns in the first iteration and the master problem has 2 rows and 90 columns in the first iteration.

In each case, the expected value problem is solved and the number of iterations, i.e., number of master problem subproblem pairs solved, is reported. Though the subproblems in the Primal-Dual decomposition are slightly larger in size than the ones in Benders decomposition, the difference in time to solve them was insignificant in all the cases. We have also reported on the number of iterations under Benders Decomposition when the upper bound has actually improved.

Empirical Results

Name	Benders		Primal-Dual
	Iterations	Improvements	
WRPM	72	28	15
HYDRO-1	119	17	10
SSN	37	23	10

Table 1

Conclusion

We have presented enhancements to the current algorithms to solve stochastic linear programs. In adapting importance-sampling techniques for stochastic linear programs, we have seen that considerable amount of computational burden can be reduced, by reducing the estimation problem from one that involves an implicit function that is expensive to evaluate, to one that is explicit and easy to evaluate. We have detailed several approaches to estimate the mean of this explicit function.

We have also demonstrated how a modified Decomposition scheme can reduce the number of iterations required to solve the expected-value problem corresponding to a stochastic linear program. We observed through numerical examples, that the reduction in computational time obtained through less number of iterations outweighs the added computational time needed by the modified method.

These enhancements give rise to faster algorithms and enable us to consider larger models which could not have been considered before. Some of the test problems we included have more than a million possible scenarios. Corresponding deterministic equivalent program will have more than a hundred million constraints. Clearly, problems of this size can not be solved by traditional mathematical programming methods.

However, problems in real life have even larger number of stochastic parameters and larger number of scenarios. The search for faster algorithms should continue in order to solve these problems. Based on our research we can suggest the following topics for future research.

The decomposition ideas presented in this dissertation are only applicable to the deterministic case. However, extending this approach to the stochastic case as well, should result in significantly large reduction in the computational time for this class. At present, a direct extension of the ideas presented in this dissertation is not possible to the stochastic case since, all the Sub problems have to have the same first stage decision in their right hand side. This makes it difficult to pass variable combinations of first stage decisions to different Sub problems since each Sub problem might choose a different convex combination. However, a further iterative loop that enforces the non-anticipativity condition may overcome this difficulty.

In sampling schemes, further research can be done to extend importance-sampling ideas to the case of dependent random variables. The case of dependent variables will not pose any difficulty in reducing the problem from integrating an implicit expensive function to integrating an explicit inexpensive function. However, newer techniques would have to be devised to integrate the resulting explicit function.

Bibliography

[1] Abrahamson P.G (1983): A Nested Decomposition Approach for Solving Staircase Line ar Programs. *Technical Report SOL 83-4, Department of Operations Research, Stanford University*

[2] Avriel M., Dantzig G.B. and Glynn P.W. (1989): Decomposition and Parallel Processing Techniques for Large Scale Electric Power System Planning Under Uncertainty. *Proceedings of the Workshop on Resource Planning under Uncertainty*

[3] Beale E.M.L: On Minimizing a Convex Function Subject to Linear Inequalities. *J.Roy.Stat.Soc. 17b, 173-184*

[4] Benders J.F. (1962): Partitioning Procedures for Solving Mixed-Variable Programming Problems. *Numerische Mathematic 4, 238-252*

[5] Birge, J.R. (1985): Decomposition and Partitioning Methods for Multi-Stage Stochastic Linear Programs. *Operations Research 33, 989-1007*

[6] Bratley P., Fox B.L., Schrage L.E.: A guide to Simulation. *Springer-Verlag 1983*

[7] Dantzig G.B. and Glynn P. (1990): Parallel Processors for Planning Under Certainty. *Annals of Operations Research 22, 1-21*

[8] Dantzig G.B (1955): Linear Programming under Uncertainty. *Management Science 1 197-206*

[9] Dantzig G.B and Wolfe P. (1960): The Decomposition Principle for Linear Programs. *Operations Research 8, 110-111*

[10] Dantzig G.B., Glynn P.W., Stone J., Entriken R., Nakayama M (1989): Decomposition Techniques for Multi-area Generation and Transmission Planning Under Uncertainty. *EPRI report 2940-1*

[11] Davis P.J. and Rabinowitz: Methods of Numerical Integration. *Academic Press, New York, 1975*

[12] Deák, I. (1988): Multidimensional Integration and Stochastic Programming, in: Ermoliev, Y. and Wets R.J. Eds.. *Numerical Techniques for Stochastic Optimization, 187-200, Springer Verlag*

[13] Dempster M.A.H (1980) (ed.): Introduction to Stochastic Programming. *Academic Press, 3-59*

[14] Edmundson H.P (1956): Bounds on the Expectation of a Convex Function for a Random Variable. *The Rand Corporation Paper 982, Santa Monica, CA*

[15] Ermoliev Y. (1988): Stochastic Quasigradient Methods, in Ermoliev Y. and Wets R.J Eds. (1988). *Numerical Techniques for Stochastic Optimization, 141-186: Springer-Verlag*

[16] Ermoliev Y. and Wets R.J Eds. (1988): Numerical Techniques for Stochastic Optimization. *Springer-Verlag*

[17] Gaivaronski A. and Nazareth L (1989): Combining Generalized Programming and Sampling Techniques for Stochastic Programs with Recourse, in. *Proceedings of the Workshop on Resource Planning under Uncertainty for Electric Power Systems, Department of Operations Research, Stanford University*

[18] Gaivoronski A. (1988): Stochastic Quasigradient Methods and their Implementation: in Ermoliev Y. and Wets R.J Eds. (1988). *Numerical Techniques for Stochastic Optimization, 313-351: Springer-Verlag*

[19] Geoffrion, A.M (1970): Elements of Large Scale Mathematical Programming. *Management Science 16, No 11, 652-675*

[20] Glynn P.W and Iglehart D. L. (1989): Importance Sampling for Stochastic Simulation. *Management Science, Vol 35, 1367-1392*

[21] Haber S.(1970): Numerical Evaluation of Multiple Integrals. *SIAM Review 12 (1970), 481-526*

[22] Higle J.L and Sen S. (1989): Stochastic Decomposition: An Algorithm for Two Stage Linear Programs with Recourse. *Mathematics of Operations Research 16/3 650-669*

[23] Infanger G. (1991): Monte Carlo (Importance) Sampling within a Benders Decomposition Algorithm for Stochastic Linear Programs, Extended Version: Including Large-Scale Results. *Technical Report SOL 91-6, Department of Operations Research, Stanford University, to appear in Annals of Operations Research*

[24] Infanger G.: Planning Under Uncertainty. *The Scientific Press*

[25] Jensen J.L : Sur les fonctions convexes et les inegalites entres les valeurs moyennes . *Acta Mathematica, 30, 175-193*

[26] Kall P. (1976): Stochastic Linear Programming. *Springer-Verlag*

[27] Madansky A.: Bounds on the Expectation of a Convex Function of a Multivariate Random Variable. *Annals of Mathematical Statistics 30, 743-746*

[28] Morton D.P. (1993): Algorithmic Advances in Stochastic Programming. *Technical Report 93-6, Systems Optimizations Laboratory, Stanford University*

[29] Murtagh B.A. and Saunders M.A.: MINOS User's Guide, SOL 82-20. *Department of Operations Research, Stanford, CA 94305*

[30] Nazareth L. and Wets R.J. (1986): Algorithms for Stochastic Programs: The Case of Non-stochastic Tenders. *Mathematical Programming Study 28, 48-62*

[31] Nazareth L. and Wets R.J. (1988): Non-linear Programming Techniques Applied to Stochastic Linear Programs with Recourse, in Ermoliev Y. and Wets R.J Eds. (1988). *Numerical Techniques for Stochastic Optimization, 95-122: Springer-Verlag*

[32] Pereira M.V, Pinto L.M.V.G., Oliveira G.C. and Cunha S.H.F (1989): A Technique for Solving LP-problems with Stochastic Right-Hand Sides. *CEPEL, Centro del Pesquisas de Energia Electria, Rio de Janeiro, Brazil*

[33] Prékopa A.: Numerical Solution of Probabilistic Constrained Programming Problem, in: Ermoliev, Y. and Wets R.J. Eds.: Numerical Techniques for Stochastic Optimization, 123-139. *Springer Verlag*

[34] Stroud A.H: Approximate Calculation of Multiple Integrals. *Prentice-Hall Englewood Cliffs, N.J., 1971*

[35] Van Slyke R.M. and Wets R.J. (1969): L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal of Applied Mathematics, Vol 17, 638-663*

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1993

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Enhanced Algorithms for Stochastic Programming

5. FUNDING NUMBERS
DE-FG03-92ER5116

6. AUTHOR(S)
Alamuru S. Krishna

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Operations Research - SOL
Stanford University
Stanford, CA 94305-4022

8. PERFORMING ORGANIZATION REPORT NUMBER
1111MA

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Energy Research
U.S. Department of Energy
Washington, DC 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SOL 93-8

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE
UL

13. ABSTRACT (Maximum 200 words)

(Abstract on back page)

14. SUBJECT TERMS
Stochastic programming, Decomposition, Monte Carlo Sampling, Stopping rules, Large-Scale

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

Abstract

In this dissertation, we present some of the recent advances made in solving two-stage stochastic linear programming problems of large size and complexity. Decomposition and sampling are two fundamental components of techniques to solve stochastic optimization problems. We describe improvements to the current techniques in both these areas.

We studied different ways of using importance sampling techniques in the context of Stochastic programming, by varying the choice of approximation functions used in this method. We have concluded that approximating the recourse function by a computationally inexpensive piecewise-linear function is highly efficient. This reduced the problem from finding the mean of a computationally expensive function to finding that of a computationally inexpensive function. Then we implemented various variance reduction techniques to estimate the mean of a piecewise-linear function. This method achieved similar variance reductions in orders of magnitude less time than, when we directly applied variance-reduction techniques directly on the given problem.

In solving a stochastic linear program, the expected value problem is usually solved before a stochastic solution is attempted. This enables us to understand the value of the stochastic solution and also to speed-up the algorithm by making use of the information obtained from the solution of the expected value problem. We have devised a new decomposition scheme to improve the convergence of this algorithm We have shown that this method gives a monotonically decreasing sequence of upper bounds whereas the regular decomposition scheme does not. We have also demonstrated that the added computational burden is insignificant compared to the reduction in the number of iterations and the resulting gain in CPU time.

DATE
FILMED
4/29/94

END