

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Augmented MCMC Simulation for Two-Stage Stochastic Programs with Recourse

Tahir Ekin

McCoy College of Business, Texas State University, *t.e18@txstate.edu*

Nicholas G. Polson

Booth School of Business, University of Chicago, *ngp@chicagobooth.edu*

Refik Soyer

School of Business, The George Washington University, *soyer@gwu.edu*

In this paper, we develop a simulation-based approach for two-stage stochastic programs with recourse. We construct an augmented probability model with stochastic shocks and decision variables. Simulating from the augmented probability model solves for the expected recourse function and the optimal first stage decision. Markov chain Monte Carlo (MCMC) methods, together with ergodic averaging, provide a framework to compute the optimal solution. We illustrate our methodology via the two stage newsvendor problem with unimodal and bimodal continuous uncertainty. Finally, we present performance comparisons of our algorithm and the sample average approximation method.

Key words: Decision analysis; Dynamic decision making; math programming; optimization; Markov chain Monte Carlo

History: Date of first-round submission: 03/07/2013; Number of revisions:3; Date of acceptance:08/13/2014

1. Introduction

Stochastic programming with recourse provides a number of computational challenges for a sequential decision maker; see Dantzig (1955). This class of decision problems is characterized by the possibility of a corrective recourse action after resolution of uncertainty. Our paper develops a simulation-based approach for computing the first and second stage optimal decisions. To do this, we define an augmented probability model over the sources of stochastic uncertainty and the decision variables. We design an MCMC algorithm with a corresponding marginal distribution on the decision that collapses on the optimal first stage decision. Our approach builds on previous literature; see Bielza et al. (1999), Müller et al. (2004), and Jacquier et al. (2007).

We focus on two-stage problems with recourse. A stage corresponds to a point of time where a set of decision variables is chosen. The decision maker computes an optimal first stage decision under uncertainty and a second stage decision after observing stochastic uncertainties. The first stage, or “here-and-now”, decision is denoted by x . The stochastic uncertainty, ξ , is drawn from a known probability distribution $p(\xi)$ which is flexible. Optimal decisions are made by maximizing the continuation expected utility. The optimal first stage decision is feasible for all scenarios of the random variable, ξ , so that the optimal expected reward (or penalty) associated with both stages is realized. After uncertainty is revealed, the second stage decision, $y(x, \xi)$, is determined by optimizing the second stage problem for the specific combination of first stage decision and the outcome value (x, ξ) . To solve this, we have the following dynamic program where $Q(x, \xi)$ is the recourse function:

$$\begin{aligned} & \max_x \quad mx - E_\xi[Q(x, \xi)] \\ & \text{subject to} \quad Ax \leq b, x \geq 0 \\ & \text{where } Q(x, \xi) = \min_y \quad q(\xi)y \\ & \text{subject to} \quad Tx + Wy \geq h, y \geq 0. \end{aligned} \tag{1}$$

At the first stage we compute \max_x rather than \min_x in the standard formulation. This notation will help us defining our augmented probability model. The second stage decision, $y(x, \xi)$, depends on the initial decision x and the source of uncertainty, ξ . The recourse function $Q(x, \xi)$ involves the minimization of $(q(\xi)y)$ subject to the second stage constraints. Here $q(\xi)$, the coefficient of second stage objective function, is a function of the random variable, ξ , and it is assumed to be integrable, $\mathbb{E}[|q(\xi)|] < \infty$. h is a deterministic parameter whereas this formulation can be extended to allow h to be a function of a random variable. The matrices W and T have appropriate dimensions conformable with x and y . The parameters m , A and b are pre-specified, deterministic arrays with appropriate dimensions.

Figure 1 illustrates the sequential nature of events. The solution involves “folding back” the decision tree while satisfying the first and second stage constraints. Specifically, if we assume $Q(x, \xi, y) = mx - q(\xi)y$, then the solution at the second decision node is given by $Q(x, \xi) = \min_y q(\xi)y$ given that y satisfies the second stage constraints. Taking the expectation with respect to the probability distribution of ξ , $p(\xi)$, at first random node and maximizing with respect to x at first decision node yields the full formulation.

1.1. Newsvendor Problem

To illustrate our methodology, we consider the classic two-stage stochastic program with recourse known as the newsvendor problem. See Dai et al. (2000), Parpas et al. (2012) and Homem-de Mello

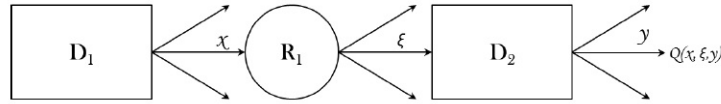


Figure 1 Decision Tree Representation of the Two-stage Recourse Problem.

and Bayraksan (2013) for the two-stage stochastic programming setup of the problem. Suppose that a newsvendor can choose the number of newspapers, x , to be purchased at stage one. We assume that $\xi = (\xi_1, \xi_2)$ represent uncertainty with demand, $d(\xi_1)$, and unit sales price, $s(\xi_2)$. The functions $d(\xi_1)$ and $s(\xi_2)$ are assumed to be non-negative. We denote the unit purchase price of a newspaper at the first stage by c and the unit resale (salvage) price at the second stage by r such that $c \geq r$. The second stage decisions on quantities of newspapers sold and salvaged are denoted by y_1 and y_2 , respectively. The newsvendor's problem is to determine x so that the overall expected profit is maximized. The problem can then be formulated as

$$\max_x -cx + E_{\xi}[Q(x, \xi)] \text{ such that } x \geq 0,$$

with $m = -c$. The recourse function, $Q(x, \xi)$ is determined by the linear program

$$Q(x, \xi) = \max_{y_1, y_2} s(\xi_2)y_1 + ry_2$$

$$\text{subject to } y_1 \leq d(\xi_1), y_1 + y_2 \leq x, y_1 \geq 0, y_2 \geq 0.$$

Once the uncertainty, ξ , is revealed, the outcomes of $d(\xi_1)$ and $s(\xi_2)$ are realized and the optimal recourse action from the linear program is given by

$$y_1^*(x, \xi) = \min(d(\xi_1), x) \text{ and } y_2^*(x, \xi) = \max(x - d(\xi_1), 0).$$

for $s(\xi_2) \geq r$, and

$$y_1^*(x, \xi) = 0 \text{ and } y_2^*(x, \xi) = x$$

for $s(\xi_2) \leq r$. The goal then is to solve the newsvendor problem with our MCMC based methodology.

The rest of the paper is outlined as follows. Section 2 defines our notation and provides a literature review. Section 3 defines our augmented probability model for stochastic programs that we use to simulate the optimal decisions and presents an MCMC algorithm using a stylized example and the newsvendor problem. Section 4 provides implementation of the augmented probability simulation (APS) model to solve the newsvendor problem for unimodal and bimodal demand distributions and compares the performance of our approach with the sample average approximation method. Finally, Section 5 concludes with directions for further research.

2. Simulation-based Stochastic Programming

The main computational difficulty in solving a two stage problem with recourse is presented by the evaluation of the expected recourse function $E_\xi[Q(x, \xi)]$ where $Q(x, \xi)$ is the optimal value function for the problem $\min_y q(\xi)y$, subject to constraints ($Tx + Wy \geq h$ and $y \geq 0$). When the stochastic uncertainty follows a discrete distribution with a small number of scenarios, the problem can be rewritten deterministically and solved via the simplex or interior point methods. A common approach for this is Benders' algorithm, see Benders (1962) and van Slyke and Wets (1969), which uses a plane cutting technique also known as the *L-shaped* method. With a discrete source of uncertainty this has the advantage of scalability. The basic idea is to represent the second stage problem by a scalar variable that cuts the objective function. The scalar variable provides an outer approximation to the expected recourse function and the cuts represent the second stage constraints in terms of first stage decision variable. A variant is the multi-cut L-shaped algorithm of Birge and Louveaux (1988).

More generally, computation becomes more difficult for continuous sources of uncertainty. Successive discrete approximation methods such as computing Jensen inequalities and Edmundson-Madansky bounds, using quadrature rules (Pennanen and Koivu (2005)) have been proposed to solve these problems as have Monte Carlo methods. Mitra (2008) provides an extensive survey of simulation based methods, hybrid methods and statistical approaches. For a recent survey, see Homem-de Mello and Bayraksan (2013).

One advantage of the Monte Carlo methods is that they are computationally tractable and applicable to both discrete and continuous sources of uncertainty. To implement direct Monte Carlo; we draw $\xi^{(g)}$ from $p(\xi)$, G times. Given this set of G sampled scenarios, we approximate the expected recourse function $E_\xi[Q(x, \xi)] = \int Q(x, \xi)P(d\xi)$ via $\hat{E}_\xi[Q(x, \xi)] = \frac{1}{G} \sum_{g=1}^G Q(x, \xi^{(g)})$.

Convergence in G is guaranteed by the law of large numbers, $\hat{E}_\xi[Q(x, \xi)] \rightarrow E_\xi[Q(x, \xi)]$, as $G \rightarrow \infty$ provided that $\mathbb{E}_\xi[|Q(x, \xi)|] < \infty$, see, for example, Dempster et al. (1977). Given this estimate of the expected recourse function, the so-called master or sample average approximation (SAA) problem is solved as a deterministic program,

$$\begin{aligned} \max_x \quad & mx - \frac{1}{G} \sum_{g=1}^G Q(x, \xi^{(g)}) \\ \text{subject to} \quad & Ax \leq b, x \geq 0 \\ \text{where } Q(x, \xi^{(g)}) = \min_y \quad & q(\xi^{(g)})y \\ \text{subject to} \quad & Tx + Wy \geq h, y \geq 0. \end{aligned} \tag{2}$$

Direct Monte Carlo methods have the advantage that they are straightforward to implement, but they can be computationally expensive in high dimensions. Various variance reduction techniques

such as antithetic variates, Latin hypercube sampling, quasi Monte Carlo, and control variates can also be used. Other alternatives are importance sampling methods, where rather than directly drawing from $p(\xi)$, one uses an importance proposal distribution, $g(\xi)$, within the L-shaped method. The proposal distribution is designed to place more weight in the region of high $Q(x, \xi)$ values, thus providing a more efficient estimate of $E_{\xi}[Q(x, \xi)]$. Proposals for $g(\xi)$ are described as additive (Infanger (1993)) or multiplicative functions (Dantzig and Thapa (1997)).

Sequential or internal sampling solution methods, where sampling is embedded within the optimization approach, have also been proposed. This strategy aims to avoid cost of the sampling effort when the current iterative solution is far from the optimal. The stochastic decomposition (SD) method of Hige and Sen (1991, 1996) uses direct Monte Carlo to estimate the expectation functions rather than computing them exactly as in regular L-shaped methods. The cuts are also updated within the algorithm as additional sample points are drawn while the algorithm proceeds. One difficulty with this approach is that the convergence to an optimum may only occur on a subsequence and that the existing cuts become redundant in the later stages of the algorithm. Regularized SD algorithms attempt to eliminate the excess cuts by adding a regularizing term as in Hige and Sen (1996). Spall (2005) provides an extensive review of the stochastic approximation algorithms that are initially considered by Rubinstein and Shapiro (1993). Ermoliev (1988) and Gaivoronski (1988) discuss the class of stochastic quasi gradient methods which are stochastic versions of sub-gradient methods designed for optimizing convex objective functions. Although all these methods can be effective in low dimensional problems, they can have poor convergence properties in high dimensions due to possible non-optimal stepsizes (Spall (2005)).

The fundamental issue in any simulation-based optimization approach is the assessment of quality of the solution and asymptotic properties of the sample estimates. If the optimal solution and objective function values are analytically available via integration, we can evaluate the quality of the solution in a straightforward manner. The difficulty arises in evaluating expectation functions analytically for many stochastic programs with continuous uncertainty, assessment of the quality of the proposed solution is not trivial and many approaches have been proposed. For example, Hige and Sen (1996) discuss stopping criteria based on duality and optimality conditions. Shapiro and Homem-de Mello (1998) develop statistical inference for the estimation of the error and propose statistically based stopping criteria for iterative algorithms. Bayraksan and Morton (2009) provide a detailed discussion and comparison of different procedures. In this paper, we use the multiple replications procedure (MRP) that is proposed by Mak et al. (1999) and revisited by Bayraksan and Morton (2009). Shapiro (2003) and Shapiro et al. (2009) discuss the theoretical properties of sampling based approaches comprehensively. Dai et al. (2000), Shapiro and Homem-de Mello (2000), Kim et al. (2011) and Polson and Sorensen (2011) also provide discussions and comparisons of the convergence rates of simulation based optimization methods.

3. An Augmented Probability Simulation Approach

In this section we develop an augmented probability model for solving the two-stage program (1.1) by maximizing the objective function

$$mx - E_{\xi}[Q(x, \xi)] \quad (3)$$

with respect to x where $Q(x, \xi) = \min_{y(x, \xi)} q(\xi)y(x, \xi)$ and $(x, y(x, \xi))$ satisfy the constraints $\{Ax \leq b, x \geq 0, Tx + Wy \geq h, y \geq 0\}$. In many cases such as the newsvendor problem, the optimal recourse action, $y^*(x, \xi)$, is available in closed form which in turn provides $Q(x, \xi) = q(\xi)y^*(x, \xi)$. In such cases finding the first stage solution x reduces to maximizing expected value of

$$u(x, \xi) = mx - q(\xi)y^*(x, \xi). \quad (4)$$

while satisfying the first stage constraints $\{Ax \leq b, x \geq 0\}$.

As proposed by Bielza et al. (1999), Müller (1999) and Müller et al. (2004), maximization of the expected value of $u(x, \xi)$ can be achieved by simulation from an augmented probability distribution $\pi(x, \xi)$ of the decision variable x and uncertainty shock ξ . Specifically, define an auxiliary distribution

$$\pi(x, \xi) = \frac{u(x, \xi)p(\xi)}{Z} \quad (5)$$

where $Z = \int u(x, \xi)p(\xi)dx d\xi$ is an appropriate normalization constant. This leads to a marginal distribution of x given by

$$\pi(x) = \frac{1}{Z} \int u(x, \xi)p(\xi)d\xi$$

which is proportional to the expected value of $u(x, \xi)$. Then the optimal value of first stage decision x is given by the mode of the marginal distribution $\pi(x)$. In the above, we assume that $u(x, \xi)$ is a nonnegative function, that is, $mx - q(\xi)y^*(x, \xi) \geq 0$ (See Jacquier et al. (2006) for transformations such as adding a positive constant to satisfy this assumption).

We simulate from the augmented probability model $\pi(x, \xi)$ using Markov chain Monte Carlo (MCMC). Specifically, we use a Gibbs sampler. This will require simulation from the conditional distribution $\pi(\xi|x)$ which is a ‘tilted’ version of the distribution $p(\xi)$ and depends on $y^*(x, \xi)$, the solution to the optimal second stage recourse function. The conditional distribution is given by $\pi(\xi|x) \propto u(x, \xi)p(\xi)$, where \propto implies that the probability density is defined up to a constant of proportionality. We also need to be able to simulate from the conditional distribution $\pi(x|\xi) \propto u(x, \xi)$.

Our approach is more efficient than direct Monte Carlo (MC) which involves first approximating $E_{\xi}[Q(x, \xi)]$ by $\hat{E}_{\xi}[Q(x, \xi)] = \frac{1}{G} \sum_{g=1}^G Q(x, \xi^{(g)})$ using G independent MC samples from $p(\xi)$ and

then optimizing $(mx - \hat{E}_\xi[Q(x, \xi)])$ over x . This is inefficient in high dimensions as the MC errors in the approximation $\hat{E}_\xi[Q(x, \xi)]$ can overwhelm the calculation of the optimal x .

The augmented probability simulation approach can be extended to include the second stage decision functions, $y(x, \xi)$, and our approach can also be used to obtain this optimal decision. In such cases, we then define the augmented joint distribution as $\pi(x, \xi, y)$ and sampling from the conditional distribution $\pi(y|x, \xi)$ will be required. Pincus (1968) provides an efficient algorithm as discussed in the next section as Algorithm 2.

3.1. Algorithm for Two-Stage Stochastic Programs

Consider the optimization problem $\min_x \{Q(x)\mathbb{I}(Ax \geq b, x \geq 0)\}$ where $Q(x)$ is continuous and attains a global unique minimum. Define a joint distribution where J denotes an algorithmic augmentation parameter

$$\pi_J(x) \triangleq \frac{1}{Z_J} \exp\{-JQ(x)\} \mathbb{I}(Ax \geq b, x \geq 0). \quad (6)$$

Here $\mathbb{I}(\cdot)$ denotes an indicator function and $Z_J = \int \exp\{-JQ(x)\} dx \mathbb{I}(Ax \geq b, x \geq 0)$ is the normalization constant. The following simulation property was initially proposed by Pincus (1968): the mode of the distribution, $\pi_J(x)$, is equal to the optimal decision, $x^* = \arg \min_x Q(x)\mathbb{I}(Ax \geq b, x \geq 0)$, and as $J \rightarrow \infty$, the mean, $E[\pi_J(x)] \rightarrow x^*$, converges to the desired mode.

For a fixed value of J , Pincus (1970) proposed a Metropolis algorithm to simulate $(x^{(0)}, x^{(1)}, \dots, x^{(g)}, \dots, x^{(G)})$ with the equilibrium distribution, $\pi_J(x)$. Under mild Harris recurrence conditions, given any starting point $x^{(0)}$, we have the limiting property $\lim_{G \rightarrow \infty} \mathbb{P}(x^{(G)} \in A | x^{(0)} = x) = \pi_J(A)$ for any Borel set A . See Tierney (1994) who ensures MCMC convergence in number of iterations, G , for $\pi_J(x)$ with a fixed value of J . Hence, if we can perform MCMC simulation of $\pi_J(x)$, we can calculate the mean as an estimator of x^* . We now provide theoretical convergence properties of our MCMC algorithm which adopts $J = J_n, n = 1, 2, \dots$ where $J_n \rightarrow \infty$ at an appropriate rate to guarantee convergence to the mode.

3.1.1. Convergence Properties: The limiting cases where $J = 0$ or ∞ are of particular interest. They both lead to uniform measures but on different sets: $J = 0$ leads to a uniform measure on the constraint set \mathcal{X} and $J = \infty$ leads to a uniform measure on the set of modes, \mathcal{X}_{min} . Specifically, with δ denoting a Dirac measure, we have

$$\lim_{J \rightarrow \infty} \pi_J(x) = \pi_\infty(x) = |\mathcal{X}_{min}|^{-1} \delta_{\mathcal{X}_{min}}(x),$$

see Geman and Geman (1987). We now let $J = J_n, n = 1, 2, \dots, \infty$, vary according to a *cooling schedule*. This creates a time inhomogeneous chain (see Isaacson and Madsen (1976)) of draws $(x^{(g)}, J^{(g)})$ that converges to $\pi_\infty(x)$ and hence finds the optimum x^* . For changing values of n ; you

get an inhomogeneous chain since the target density changes with n . We will see that we cannot increase J_n too fast, as probably we can get stuck in a local mode. General convergence properties of an inhomogeneous Markov chain follow from Müller et al. (2004).

Our augmented probability model is defined by using J *iid* draws of ξ , $\xi_J = (\xi_1, \dots, \xi_J)$ where we assume bounded utility (objective) functions; $u_0 \leq u(x, \xi_j) \leq u_1$ for all J . The distribution of interest, $\pi_J(x)$, is the marginal implied from the joint distribution

$$\pi_J(x, \xi_J) \propto \prod_{j=1}^J [u(x, \xi_j) p(\xi_j | x)]. \quad (7)$$

Equivalently, let $v = \frac{1}{J} \sum_{j=1}^J \log u(x, \xi_j)$ be the average of log utility functions. Define the transition probabilities; $P_{ij}(n) = P(x_{n+1} = x', v_{n+1} = v' | x_n = x, v_n = v)$ on the space of (x, v) for $i = (x, v)$ and $j = (x', v')$ with the stationary distribution $\pi_J^*(x, v)$. We assume there exist constants, $0 < a < \infty$, $0 \leq b < \infty$ and assume $p(v|x) \geq \epsilon > 0$ for all (x, v) , so that $p_J(v|x) \geq \epsilon^J$ and that the proposal density, $p(x'|x) \geq a$. This allows us to have a bound on the transition kernel, $P_{ij}(n)$, which is related to the coefficient of ergodicity

$$\tau[P(n)] = 1 - \inf_{i,j} \sum_{l=1}^m \inf(P_{il}(n), P_{jl}(n)).$$

The convergence rate is characterized by

$$\delta[P(n)] = 1 - \tau[P(n)],$$

which is referred to as the delta coefficient of $P(n)$, see Isaacson and Madsen (1976).

As pointed out by Müller et al. (2004), to show weak ergodicity to the desired stationary distribution, $\pi_\infty^*(x, v)$, we pick J_n to ensure that $\sum_{n=0}^{\infty} \{1 - \tau[P(n)]\} = \infty$ for our sequence of stationary distributions. For a fixed J_n , a bound is also established, $P_{ij}(n) \geq a \exp\{-bJ_n\}$, that results with geometric convergence with a rate characterized by $\delta[P(n)]$. By setting $b = \log(u_1/u_0\epsilon)$ and using the assumptions on the utility, we have

$$P_{ij}(n) \geq a(u_0\epsilon/u_1)^{J_n} = a \exp\{-bJ_n\}.$$

If we pick the *cooling schedule*, $J_n = \frac{1}{\gamma} \log(n+c)$, then

$$\sum_{n=0}^{\infty} [1 - \tau(P(n))] \geq a \sum_{n=0}^{\infty} \exp\{-bJ_n\} = a \sum_{n=0}^{\infty} 1/(n+c)^{b/\gamma} = \infty$$

which holds for $c > 0$ and $\gamma \geq b$. Therefore we have weak ergodicity of our chain. Strong ergodicity follows from the Theorem A.1 of Müller et al. (2004).

For practical purposes, instead of a formal implementation we rather recommend a diagnostic approach that increases $J^{(g)}$ until the MCMC draws stabilise (see Gelman and Rubin (1992)). The geometric rate of convergence depends on b , a function of the bounds of the utility, u_0 and u_1 . Practically, these bounds can be chosen with respect to the maximum and minimum values of the utility function, $\max Q(x)$ and $\min Q(x)$. Then, it is enough to sample from the joint density and then calculate the ergodic mean $\frac{1}{G} \sum_{g=1}^G X^{(g)}$ along the chain for a value of J on a given schedule. Figure 2, that is presented in Section 4, presents evidence for numerical convergence rate which is geometric for increasing values of J . Practical convergence in J is also assessed for reasonable values of J such as $J = 5, 10, 25, 50, 100$ in most applications (Jacquier et al. (2007)). For other convergence asymptotics see Dai et al. (2000), Shapiro and Homem-de Mello (2000), and Polson and Sorensen (2011).

3.1.2. Modified Augmented Probability Model: Given these convergence results, we modify our APS by defining the augmented probability model with J *iid* draws of ξ as

$$\pi_J(x, \boldsymbol{\xi}_J) \propto \prod_{j=1}^J \left[(mx - q(\xi_j)y_j)p(\xi_j)\mathbb{I}(Ax \leq b, x \geq 0, Tx + Wy_j \geq h, y_j \geq 0) \right], \quad (8)$$

where for each x , J unique draws of (ξ_j) are generated and $y_j^* = y^*(x, \xi_j)$ are computed for each pair. We assume that $(mx - q(\xi_j)y_j^*) \geq 0$ for all J terms, $\boldsymbol{\xi}_J = (\xi_1, \dots, \xi_J)$ are J independent draws from $p(\xi)$.

The key feature of this distribution is that the mean of its marginal distribution $\pi_J(x)$ for the first stage variable converges to the optimum for a sufficiently large enough value of J . Since ξ_j 's are *iid* draws, the marginal distribution is given by

$$\pi_J(x) \propto E_\xi \left[(mx - q(\xi)y) \mathbb{I}(Ax \leq b, x \geq 0, Tx + Wy \geq h, y \geq 0) \right]^J$$

where

$$E_\xi [(mx - q(\xi)y)]^J = \left[\int (mx - q(\xi)y)p(\xi)d\xi \right]^J.$$

This corresponds to our previous discussion. The marginal $\pi_J(x) \propto e^{Q(x)}$ where

$$Q(x) = \ln E_\xi \left[(mx - q(\xi)y) \mathbb{I}(Ax \leq b, x \geq 0, Tx + Wy \geq h, y \geq 0) \right]^J.$$

Therefore by computing the mean of the samples from $\pi_J(x)$, we can find the optimal first stage decision x^* .

To sample from $\pi_J(x, \boldsymbol{\xi}_J)$ we use MCMC. A Gibbs sampler can be constructed using conditional distributions $\pi_J(x|\boldsymbol{\xi}_J)$ and $\pi_J(\boldsymbol{\xi}_J|x)$. Samples from the joint conditional $\pi_J(\boldsymbol{\xi}_J|x)$ can be obtained

by independent draws from the conditionals $\pi_J(\xi_j|x)$ for $j = 1, \dots, J$. Specifically, for $j = 1, \dots, J$, we use

$$\begin{aligned}\pi_J(x|\boldsymbol{\xi}_J) &\propto \prod_{j=1}^J (mx - q(\xi_j)y_j^*) \mathbb{I}(Ax \leq b, x \geq 0) \\ \pi_J(\xi_j|x) &\propto (mx - q(\xi_j)y_j^*)p(\xi_j).\end{aligned}\tag{9}$$

By starting at initial values $(x^{(0)}, \boldsymbol{\xi}_J^{(0)})$ and sampling through the conditionals we define a Markov chain where $(x^{(g)}, \boldsymbol{\xi}_J^{(g)}) \xrightarrow{D} (x^{(\infty)}, \boldsymbol{\xi}_J^{(\infty)})$ as $g \rightarrow \infty$, with the pair $(x^{(\infty)}, \boldsymbol{\xi}_J^{(\infty)})$ having distribution $\pi_J(x, \boldsymbol{\xi}_J)$ where \xrightarrow{D} represents convergence in distribution. Then the marginal draws $x^{(g)}$ will have distribution $\pi_J(x)$. Hence, we can use the Monte Carlo average of our draws to estimate the mean of $\pi_J(x)$, which in turn provides us an estimate of x^* . One of the advantages of jointly simulating decision variable x and sources of uncertainty ξ is that we take draws with high objective function values.

In many cases, the full conditional distributions may not be standard distributions and in such cases, as proposed in Bielza et al. (1999), we use the Metropolis algorithm to draw samples. During the implementation of the Metropolis algorithm, the choice of the proposal distribution requires that we have an irreducible and aperiodic Markov Chain. A standard choice is a symmetric distribution such as normal or heavy tailed t with appropriate scaling. Bielza et al. (1999) provides a further discussion regarding the specification of the proposal distribution. The details of our algorithm can now be summarized as follows:

3.1.3. Algorithm 1: Here the augmented probability space is constructed on $(x, \boldsymbol{\xi}_J)$. We assume that the optimal recourse action $y_j^*(x, \xi_j)$ for each j^{th} scenario is available (or can be obtained numerically) from the second stage as in the Newsvendor problem.

1. Start with initial values: $(x^{(0)}, \xi_j^{(0)}, j = 1, \dots, J)$, where $x^{(0)}$ satisfies the constraints $\{x|Ax^{(0)} \leq b, x^{(0)} \geq 0\}$. Set $i = 1$.
2. Compute $y_j^*(x^{(i-1)}, \xi_j^{(i-1)})$ for all J scenarios.
3. Let $u_1(x^{(i)}, \xi_j^{(i-1)}) = mx^{(i)} - q(\xi_j^{(i-1)})y_j(x^{(i)}, \xi_j^{(i-1)})$ and draw $x^{(i)}$ from the full conditional distribution

$$\pi(x^{(i)}|\boldsymbol{\xi}_J^{(i-1)}) \propto \prod_{j=1}^J u_1(x^{(i)}, \xi_j^{(i-1)}) \mathbb{I}(Ax^{(i)} \leq b, x^{(i)} \geq 0).$$

To draw $x^{(i)}$, generate a candidate $x_*^{(i)}$ from the proposal distribution $g_1(x_*^{(i)}|x^{(i-1)})$ and evaluate $u_1(x_*^{(i)}, \boldsymbol{\xi}_J^{(i-1)})$.

Assuming g_1 is a symmetric proposal density, compute

$$a_1 = \min\left\{1, \prod_{j=1}^J \frac{u_1(x_*^{(i)}, \xi_j^{(i-1)})}{u_1(x^{(i)}, \xi_j^{(i-1)})}\right\}.$$

With probability a_1 , set $x^{(i)} = x_*^{(i)}$ otherwise, keep $x^{(i)} = x^{(i-1)}$.

4. Let $u_2(x^{(i)}, \xi_j^{(i)}) = mx^{(i)} - q(\xi_j^{(i)})y_j^*(x^{(i)}, \xi_j^{(i)})$. Draw $\xi_j^{(i)}$ for $j = 1, \dots, J$ from the full conditional distribution

$$\pi(\xi_j^{(i)} | x^{(i)}) \propto u_2(x^{(i)}, \xi_j^{(i)})p(\xi_j^{(i)}).$$

To draw $\xi_j^{(i)}$, generate a candidate $\xi_{j*}^{(i)}$ from the proposal distribution $g_2(\xi_{j*}^{(i)} | x^{(i)})$ and evaluate $u_2(x^{(i)}, \xi_{j*}^{(i)})$.

Assuming g_2 is a symmetric proposal density, compute

$$a_2 = \min \left\{ 1, \frac{u_2(x^{(i)}, \xi_{j*}^{(i)})}{u_2(x^{(i)}, \xi_j^{(i)})} \right\}.$$

With probability a_2 , set $\xi_j^{(i)} = \xi_{j*}^{(i)}$ otherwise, keep $\xi_j^{(i)} = \xi_j^{(i-1)}$.

5. Set $i = i + 1$. Repeat steps 2 through 4 until the Markov chain is judged to have practically converged.

In implementation of the algorithm, we check for convergence at every so many iterations. When convergence is achieved we calculate the mean (or mode) of x and compute the optimality gap. If the mean of x is feasible for all scenarios in the second stage and the optimality gap is smaller than the tolerance level, we stop and decide that this value is the estimate of the optimal decision. To evaluate the optimality gap we use the multiple replications procedure (MRP) of Bayraksan and Morton (2009) that will be discussed in the next section. If the optimality gap is not smaller than the tolerance, then our strategy is to increase the value of J and to rerun the algorithm as to be implemented in Section 4.

3.1.4. Algorithm 2: When the optimal recourse action $y^*(x, \xi)$ is not available in closed form, the augmented probability model can be written as

$$\pi_J(x, \xi_J, \mathbf{y}_J) \propto \prod_{j=1}^J \left[\left(mx - q(\xi_j)y_j \right) p(\xi_j) \mathbb{I}(Ax \leq b, x \geq 0, Tx + Wy_j \geq h, y_j \geq 0) \right] \quad (10)$$

where $\mathbf{y}_J = (y_1, \dots, y_J)$.

We need to draw samples from $\pi_J(x, \xi_J, \mathbf{y}_J)$ using the Gibbs sampler, and in turn this requires samples from the conditional distribution $\pi(\mathbf{y}_J | x, \xi_J)$. Draws from this joint conditional distribution can be obtained from the independent conditionals $\pi(y_j | x, \xi_j)$ for $j = 1, \dots, J$. We can recognize that $y_j^* = y(x, \xi_j)$ is the optimal solution of the minimization problem at the second stage under scenario ξ_j . Specifically, y_j^* is the minimum of the problem $\min_y q(\xi_j)y_j$ subject to $(Tx + Wy_j \geq h, y_j \geq 0)$ and, following Pincus (1968) we can use the conditional distribution of y_j as

$$\pi(y_j | x, \xi_j) \propto \exp \{ -Jq(\xi_j)y_j \} \mathbb{I}(Tx + Wy_j \geq h, y_j \geq 0), \text{ for } j = 1, \dots, J. \quad (11)$$

Algorithm 1, then needs to be modified to incorporate drawing samples from the conditional distribution of y_j 's. The modified algorithm is discussed in the Appendix and an implementation is presented.

3.2. Related Literature

Our approach for finding the first stage solution x and maximizing $u(x) = E_{\xi}[u(x, \xi)]$ without resorting to gradient methods is related to the simulated annealing algorithm. Pure simulated annealing also powers up a function to be maximized in order to find its optimum (Kirkpatrick et al. (1983)). Aarts and Korst (1988) provide a detailed discussion about use of simulated annealing for optimization. Pincus (1968) proposes a solution for deterministic optimization problems by analytically computing the ratio of annealed integrals. However, as noted by Müller et al. (2004), standard simulated annealing algorithms (see van Laarhoven and Aarts (1987) for a review) assume availability of $u(x)$ for direct evaluation.

Pincus (1970) presents the use of Metropolis algorithm to approximate the ratio of integrals. Similarly, APS evaluates $u(x)$ in an iterative manner that simultaneously address maximization and integration by using MCMC. When the utility (objective) function $u(x)$ can be evaluated exactly, APS chain reduces to a standard simulated annealing algorithm as described by Müller (1999). Bielza et al. (1999) utilize APS to solve decision problems and describe the procedure while fixing $J = 1$. Müller et al. (2004) extend it to the solution of decision problems where the utility function may not be evaluated exactly and might have complex, high dimensional decision spaces by using higher values of the augmentation parameter, J . Similar ideas have been utilized to solve maximization of expected utility problem (Jacquier et al. (2006)), to compute maximum likelihood estimates in latent state variable models (Jacquier et al. (2007)) and to solve stochastic dynamic programs (Polson and Sorensen (2011)).

Our first algorithm extends the proposed APS approach to stochastic programs, particularly to two stage stochastic programs with recourse within constrained domains which is our main difference from the approach of Müller et al. (2004). We also provide performance comparisons with Sample Average Approximation method that is widely used in stochastic programming literature. In addition to that, Algorithm 2 (See Appendix) utilizes the deterministic optimization approach of Pincus (1968) to provide distributional form of the second stage decision in a two stage sequential decision setting. In doing that, our paper provides a connection between Pincus (1968) and APS algorithm of Bielza et al. (1999). The latter did not consider constrained domains and the former dealt with only one stage problems where the objective function was not analytically available.

3.3. APS Formulation of the Newsvendor Problem

In this section, we consider the two-stage stochastic programming formulation of the Newsvendor problem presented in Section 1.1. More specifically, we have

$$\max_x -cx + E_{\xi}[Q(x, \xi)] \text{ such that } x \geq 0,$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2)$, and $Q(x, \boldsymbol{\xi}) = s(\xi_2)y_1^*(x, \boldsymbol{\xi}) + ry_2^*(x, \boldsymbol{\xi})$. The optimal recourse actions are given by $y_1^*(x, \boldsymbol{\xi}) = \min(d(\xi_1), x)$, $y_2^*(x, \boldsymbol{\xi}) = \max(x - d(\xi_1), 0)$ for $s(\xi_2) \geq r$, and $y_1^*(x, \boldsymbol{\xi}) = 0$, $y_2^*(x, \boldsymbol{\xi}) = x$ for $s(\xi_2) \leq r$.

Following our development in the above, using J *iid* copies of the random vector (ξ_1, ξ_2) , we can write down the augmented distribution, $\pi_J(x, \boldsymbol{\xi}_{1J}, \boldsymbol{\xi}_{2J})$ as

$$\pi_J(x, \boldsymbol{\xi}_{1J}, \boldsymbol{\xi}_{2J}) \propto \prod_{j=1}^J (-cx + s(\xi_{2j})y_{1j}^* + ry_{2j}^*)p(\xi_{1j}, \xi_{2j}),$$

where $\xi_{i,j}$ is the j^{th} draw of ξ_i , $\boldsymbol{\xi}_{iJ} = \{\xi_{i,1}, \dots, \xi_{i,J}\}$ for $i = 1, 2$, $\boldsymbol{\xi}_j = \{\xi_{1j}, \xi_{2j}\}$, $y_{1j}^* = y_1^*(x, \boldsymbol{\xi}_j)$ and $y_{2j}^* = y_2^*(x, \boldsymbol{\xi}_j)$.

We use Gibbs sampling and draw from the full conditional distributions; $\pi_J(x|\boldsymbol{\xi}_{1J}, \boldsymbol{\xi}_{2J})$ and $\pi_J(\xi_{1j}, \xi_{2j}|x)$ for $j = 1, \dots, J$. If ξ_{1j} and ξ_{2j} are independent, we can sample from their marginal conditionals. However, for any choice of the $p(\xi_{1j}, \xi_{2j})$, one can not directly draw samples from these distributions. We use a Metropolis algorithm as presented by the Algorithm 1 using Gaussian proposal distributions in both cases.

In Section 4 we assume the functions $d(\xi_{1j}) = c_1 \exp(\xi_{1j})$ and $s(\xi_{2j}) = c_2 \exp(\xi_{2j})$. In this case it can be shown that the full conditionals of ξ_{1j} and ξ_{2j} are both mixtures which still requires use of the Metropolis method.

As pointed out by Parpas et al. (2012), samples from multi-modal distributions often preclude the application of stochastic programming. Our realizations of the random variable being dependent on the incumbent candidate solution and the use of J realizations in each iteration makes our algorithm a viable candidate for dealing with multi-modal distributions. In Section 4 we consider the newsvendor problem with a bi-modal demand distribution. More specifically, we assume that $d(\xi_1)$'s distribution is a mixture of two gamma densities.

4. Performance of the APS Algorithm

In this section, we evaluate performance of APS algorithm using the newsvendor problem and compare it with the Sample Average Approximation (SAA) algorithm. SAA algorithm is chosen because it was adopted as the benchmark in similar influential related works (Higle and Sen (1996)) and it has been widely used to solve similar stochastic programs (Birge and Louveaux (2011)). Evaluation of optimality gap plays an important role in our assessment of the algorithm's performance. In so doing, we use the multiple replications procedure (MRP) of Mak et al. (1999) and Bayraksan and Morton (2009) to assess the quality of the candidate solution. In what follows, we give an overview of the MRP.

4.1. Assessment of the optimality gap

To assess the optimality, the optimal objective function value is either computed by using the optimal solution, x^* or is estimated by using a Monte Carlo average (as in SAA) when x^* is not analytically available. The optimality gap of a candidate solution, x_* , is found using n_g replications with sample size n in each case. A summary of MRP procedure is provided below:

- For every replication, $k = 1, 2, \dots, n_g$
 - Sample n *iid* observations $\xi^k = \{\xi^{k1}, \xi^{k2}, \dots, \xi^{kn}\}$ from $p(\xi)$
 - Solve the stochastic program for the sample of ξ^k to compute x_k^* , an estimate for the optimal solution x^*
 - Compute the optimal objective function value for the sample ξ^k as $Q(x_k^*, \xi^k)$
 - Compute the objective function value for the candidate solution, x_* for the sample ξ^k as $Q(x_*, \xi^k)$

- Compute the gap estimate, $D_n^k(x_*) = \frac{1}{n} \sum_{j=1}^n [|Q(x_*, \xi^{kj}) - Q(x_k^*, \xi^{kj})|]$.

- Compute the average and sample variance of the gap estimate as

$$\bar{D}_{n_g}(x_*) = \frac{\sum_{k=1}^{n_g} D_n^k(x_*)}{n_g}, \text{ and } s_{n_g}^2(x_*) = \frac{1}{n_g - 1} \sum_{k=1}^{n_g} (D_n^k(x_*) - \bar{D}_{n_g}(x_*))^2.$$

For large values of n_g , due to the central limit theorem, confidence interval for the optimality gap can also be obtained.

4.2. Performance Results

In this section, we present the solution of the Newsvendor problem with uni-modal and bi-modal demand distributions using the APS algorithm with different augmentation parameter values, J , and Sample Average Approximation (SAA) algorithm. We present evidence of the effectiveness of APS versus SAA for both cases.

We first consider the uni-modal case with $d(\xi_1) = c_1 \exp(\xi_1)$ and $s(\xi_2) = c_2 \exp(\xi_2)$, as the demand and sales price functions, respectively. As defined before, the random vector $\xi = \{\xi_1, \xi_2\}$ follows normal distribution that results with demand and sales price functions having lognormal distributions. We assume that $c_1 = 100$, $c_2 = 1.5$ and ξ_1 and ξ_2 are independent normally distributed random variables with $\mu_i = 0$ and $\sigma_i^2 = 1$ for $i = 1, 2$. Furthermore, the unit cost and the salvage values are set as $c = 1$ and $r = 0.1$, respectively. In this particular case, the actual optimal solution x^* can be computed numerically.

The APS is run with J values of 25, 50 and 100. Sample Average Approximation (SAA) is run for 25 replications. Candidate solutions are computed every 100 iterations. We draw a sample of ξ with size 150,000 to estimate the continuous distribution, $p(\xi)$. Optimality gap is computed using

Mean Absolute Percentage Error (MAPE). It is a function of the difference between the objective function value for the candidate solution of a particular approach and objective function value for the optimal solution that are both computed using the same sample with 150,000 realizations.

Figure 2 presents numerical evidence of geometric convergence by providing optimality gaps for J values of $\{10, 25, 50, 100\}$ and cases of 10,000, 20,000, 30,000, 40,000 realizations where number of realizations is given by $(J \times G)$ where G is the number of MCMC iterations in each case.

In Figure 3, we present the MAPE for both SAA and APS algorithms. The results are presented for increments of 5000 realizations for the SAA and the APS with $J = 25$ and with $J = 50$. For the case of APS with $J = 100$ we present the results for increments of 10,000. For example, in the case of APS with $J = 100$ sample of 10,000 implies $G = 100$ MCMC draws. It is important to note that Figure 3 does not show the MAPE for the APS with $J = 25$ and with $J = 50$ for 10,000 realizations. This is due to the lack of convergence of the MCMC algorithm with $G = 400$ draws for $J = 25$ and $G = 200$ draws for $J = 50$. The convergence of the MCMC algorithm was checked using the BGR statistic; see for example, Brooks and Roberts (1997). For 20000 realizations, the MCMC chains are judged to have practically converged using evidence of BGR values of 1.020, 1.021, 1.027 for simulations with $J = 25$, $J = 50$ and $J = 100$, respectively. We note that the APS with $J = 100$ performs better than the SAA algorithm at any sample size. Even the APS with $J = 25$, becomes superior than SAA algorithm with 20,000 realizations. The SAA recovers and performs as good or better than the APS with $J = 25, J = 50$ for larger number of realizations, but it does not perform as good as the APS algorithm with $J = 100$.

We provide the box-plots of optimality gaps for SAA and APS with $J = 25, J = 100$ for 20,000, 30,000, 40,000 and 50,000 realizations (see Figure 4) to understand the evolution of the performance and provide further comparisons. Box-plots of APS with $J = 25$ do not exhibit any significant performance increase in mean optimality gap but results with reduced variance for increased sample sizes compared to the SAA. However, APS with $J = 100$ provides smaller mean optimality gaps as well as smaller standard deviation values for any sample size.

Figure 5 elaborates on the visualization of the performance comparisons for a given sample size by presenting the box-plots of the optimality gaps for all the approaches with 30,000 realizations. It can be seen that APS runs result with better mean absolute percentage optimality gaps whereas their standard deviations are also smaller. Among the APS runs, APS with larger values of J provides better performance. Figure 6 provides the box-plots of the first stage decision variable, x , compared with the optimal solution, x^* . The standard deviation of the draws for a higher value of J is smaller which presents evidence for faster convergence.

Overall, practical simulation runs are in line with our expectations from the theoretical results. The APS algorithm converges faster and performs consistently better than the SAA as for large

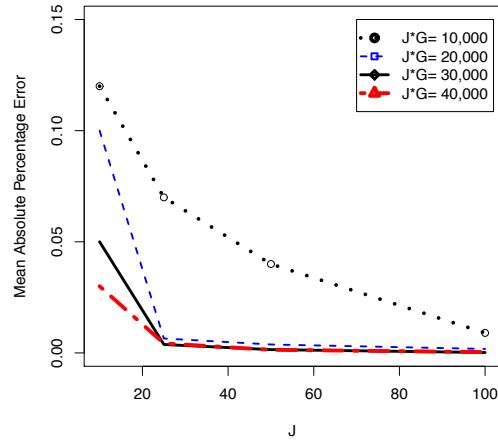


Figure 2 Optimality gaps for different values of J

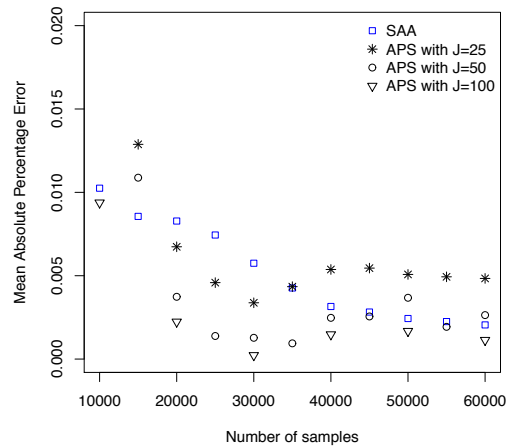


Figure 3 Optimality gap for SAA and APS with the same unimodal demand sample

values of J . Thus, as an implementation strategy, once the MCMC convergence is achieved it is better to increase J rather than the number of MCMC iterations G to improve the optimality gap of the APS algorithm.

Another analysis is done to compare the samples of the random vector, ξ drawn by SAA and APS methods. An expected advantage of the APS approach is smarter sampling from the state space that leads to better objective function values. In Figure 7, we present the comparison of SAA and the APS with $J = 25$. Similar results can be obtained for other values of J . We have used the samples of ξ that are drawn from each particular algorithm rather than the common independent

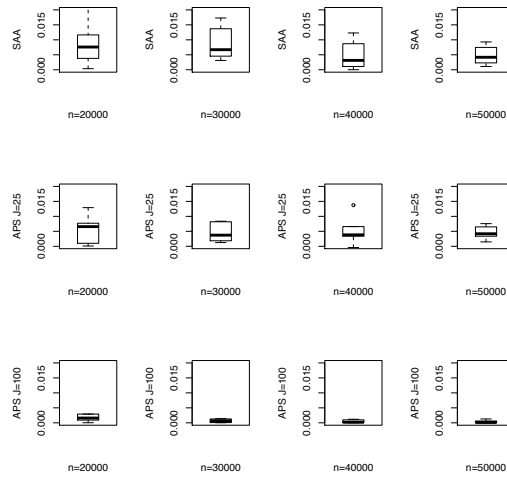


Figure 4 Box-plots of the optimality gap with the same unimodal demand samples

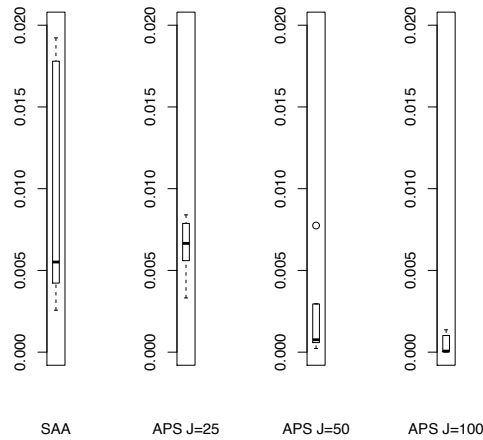


Figure 5 Box-plots of the optimality gap for SAA and APS with the same unimodal demand sample of 30,000 draws

sample. Objective function values are computed for the candidate solutions using the own samples of each algorithm and compared to the optimal objective function value that is computed for the exact optimal solution based on a large sample of 150,000 realizations. As we can see from Figure 7, the standard deviations of the objective functions are smaller for APS samples which are in line with the importance sampling results of Parpas et al. (2012) and Infanger (1993). Furthermore, based on the MAPE, the optimality gap for APS approach is also consistently found to be smaller compared to SAA approach.

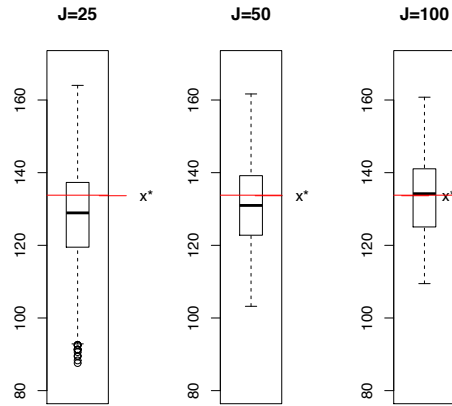


Figure 6 Box-plots of the first stage decision, x , for $J = 25$, $J = 50$ and $J = 100$ with the same unimodal demand sample

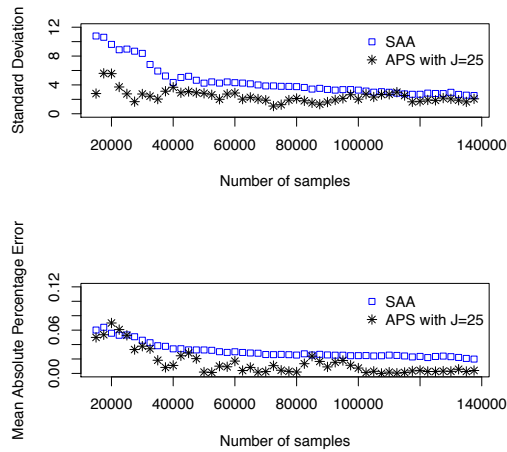


Figure 7 Standard deviation and optimality gap for SAA and APS with own unimodal demand samples

Next, we consider the newsvendor problem with a bi-modal demand distribution. Specifically, we assume that distribution of $d(\xi_1)$ is a mixture of gamma densities as

$$p(d(\xi_1)) = 0.5 \text{Gamma}(a = 150, b = 1) + 0.5 \text{Gamma}(a = 150, b = 3),$$

where $\text{Gamma}(a, b)$ denotes a gamma density with shape parameter a and scale parameter b . We have $s(\xi_2) = c_2 \exp(\xi_2)$ with a c_2 value of 1.5 and ξ_2 is a normally distributed random variable with mean, $\mu_2 = 0$ and variance, $\sigma_2^2 = 1$.

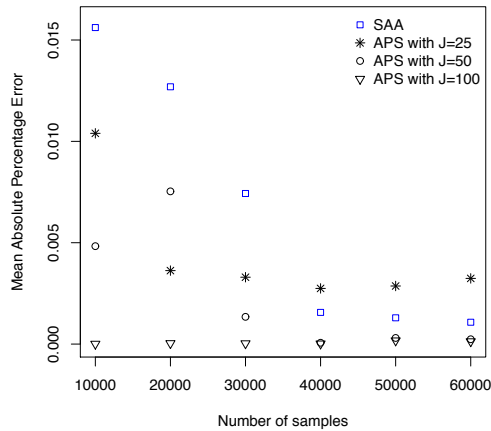


Figure 8 Optimality gap for SAA and APS with the same unimodal demand sample

In this case, the optimal solution can not be computed easily via numerical methods. Thus, we have found the optimal solution and optimal objective function value using Sample Average Approximation for a large enough sample of 150,000 realizations. This is used to estimate the optimal solution of the original problem.

The APS algorithm was implemented with J values of 25, 50, 100. The SAA algorithm was run for 25 replications. We will not provide a detailed analysis and figures to keep the paper parsimonious. The MAPEs for the optimality gaps are reported in Figure 8. As can be clearly seen from the figure, APS with J values of 50 and 100 are found to be superior for all number of samples. Box-plot of the first stage decision variable, x , for the first 20000 iterations also provides similar insights (See Figure 9). Increasing J value results with a smaller error from the optimal solution and a smaller standard deviation.

Similar to the analysis with uni-modal demand, APS samples are found to be superior than the SAA Monte Carlo samples as shown in Figure 10. APS provides objective function values with smaller standard deviation and smaller optimality gap for all sample sizes. Overall, the efficiency of APS can be concluded to increase especially when used with its own samples and large enough J values.

The performance results are generally robust with respect to increasing standard deviation of the random variables. As expected, for major increases of the standard deviation, both our algorithm as well as the SAA algorithm result with higher optimality gaps and variances when run with the same number of samples.

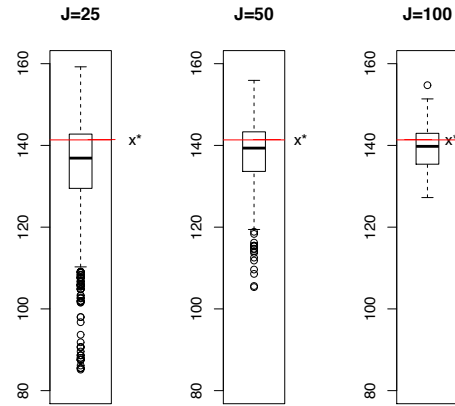


Figure 9 Box-plots of the first stage decision, x , for $J = 25$, $J = 50$ and $J = 100$ with the same unimodal demand sample

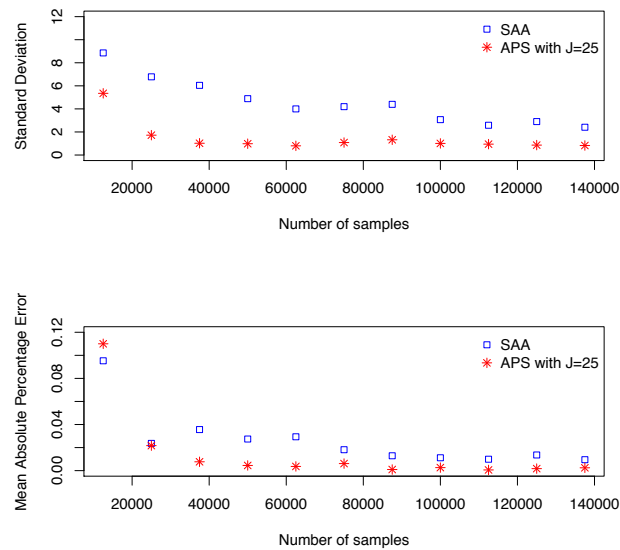


Figure 10 Standard deviation and optimality gap for SAA and APS with own samples of bi-modal demand

5. Discussion

Solving two-stage stochastic programming with recourse requires optimization and calculation of expected recourse function, $Q(x)$. In this paper, simulation from the proposed augmented probability model where both decision variables and uncertainty shocks are treated as stochastic, allow us to perform both simultaneously. In doing that, we sample more frequently from the augmented

decision and state space in that the objective function has higher values. This avoids inefficient samples that otherwise would have been drawn without considering the optimization perspective. We also avoid having to use gradient-based approaches.

For performance comparison, the newsvendor problem is formulated as a two-stage stochastic program with recourse and solved by both the APS and SAA algorithms. It is shown that APS provides smaller optimality gaps for the same number of samples compared to SAA. APS also results with samples with reduced variance. In an APS algorithm, faster convergence can be achieved by increasing J .

Markov chain Monte Carlo (MCMC) methods are utilized to sample from resulting non-standard distributions and compute the optimal solution. We use Metropolis algorithm to sample from the complete conditionals. However, sampling may not be straightforward and the convergence may not be fast enough for some specific cases. When the objective function is flat and the optimal decision is not significantly better than the other alternatives, the mode of the pseudo distribution may not be distinctive. Increasing the value of the augmentation parameter, J , resolves this problem since the mean converges to the optimal solution for high values of J .

During the implementation of the algorithm, user may need to deal with certain issues. Firstly, the objective function is assumed to be positive within the APS algorithm. For negative objective functions, certain transformations can be used to achieve nonnegativity (Jacquier et al. (2006)). Secondly, sampling from constrained domains with a MCMC algorithm remains a challenge. The use of Lagrangian penalty parameters can be considered.

There are a number of directions for future research. The proposed approach can be extended to solve other multi-stage stochastic problems. It may be worthwhile to explore how it applies to various non-linear stochastic programming problems and dynamic stochastic programs. Another interesting area for work is recourse problems where probability distribution of source of uncertainty is dependent on the first stage decision variable, see for example, Jonsbraten et al. (1998).

Acknowledgments

We thank the editors and two anonymous reviewers whose comments helped improve and clarify this manuscript.

Appendix. Algorithm 2

In cases where the optimal recourse action is not easy to evaluate, the Algorithm 1 of Section 3.1 needs to be modified. In this case, the augmented probability space is constructed on $(x, \boldsymbol{\xi}_J, \mathbf{y}_J)$ and the full conditional distribution of y_j 's can be constructed using the Pincus (1968) result. The details of our algorithm can be summarized as follows:

1. Start with initial values: $(x^{(0)}, \xi_j^{(0)}, y_j^{(0)}, j = 1, \dots, J)$, where $x^{(0)} \in S = \{Ax^{(0)} \leq b, x^{(0)} \geq 0, y_j^{(0)} \geq 0, Tx^{(0)} + Wy_j^{(0)} \geq h\}$. Set $i = 1$.
2. Let $u_1(x^{(i)}, \xi_j^{(i-1)}, y_j^{(i-1)}) = mx^{(i)} - q(\xi_j^{(i-1)})y_j^{(i-1)}$ and draw $x^{(i)}$ from the full conditional distribution

$$\pi(x^{(i)} | \boldsymbol{\xi}_J^{(i-1)}, \mathbf{y}_J^{(i-1)}) \propto \prod_{j=1}^J (u_1(x^{(i)}, \xi_j^{(i-1)}, y_j^{(i-1)}) \mathbb{I}(Ax^{(i)} \leq b, x^{(i)} \geq 0)).$$

While using MCMC, generate a candidate $x_*^{(i)} \sim g_1(x_*^{(i)} | x^{(i-1)})$ and evaluate $u_1(x_*^{(i)}, \xi_j^{(i-1)}, y_j^{(i-1)})$, where g_1 is a symmetric proposal density. Compute

$$a_1 = \min\left\{1, \prod_{j=1}^J \frac{u_1(x_*^{(i)}, \xi_j^{(i-1)}, y_j^{(i-1)})}{u_1(x^{(i)}, \xi_j^{(i-1)}, y_j^{(i-1)})}\right\}.$$

With probability a_1 , set $x^{(i)} = x_*^{(i)}$ otherwise, keep $x^{(i)} = x^{(i-1)}$.

3. Let $u_2(x^{(i)}, \xi_j^{(i)}, y_j^{(i-1)}) = mx^{(i)} - q(\xi_j^{(i)})y_j^{(i-1)}$. Draw $\xi_j^{(i)}$ for $j = 1, 2, \dots, J$ from the full conditional distribution

$$\pi(\xi_j^{(i)} | x^{(i)}, y_j^{(i-1)}) \propto (u_2(x^{(i)}, \xi_j^{(i)}, y_j^{(i-1)}))p(\xi_j^{(i)}).$$

In so doing, generate a candidate $\xi_{j*}^{(i)} \sim g_2(\xi_{j*}^{(i)} | x^{(i)}, y_j^{(i-1)})$ from the symmetric proposal density g_2 and evaluate $u_2(x^{(i)}, \xi_{j*}^{(i)}, y_j^{(i-1)})$. Compute

$$a_2 = \min\left\{1, \frac{u_2(x^{(i)}, \xi_{j*}^{(i)}, y_j^{(i-1)})}{u_2(x^{(i)}, \xi_j^{(i)}, y_j^{(i-1)})}\right\}.$$

With probability a_2 , set $\xi_j^{(i)} = \xi_{j*}^{(i)}$ otherwise, keep $\xi_j^{(i)} = \xi_j^{(i-1)}$.

4. Let $u_3(x^{(i)}, \xi_j^{(i)}, y_j^{(i)}) = mx^{(i)} - q(\xi_j^{(i)})y_j^{(i)}$. Draw $y_j^{(i)}$ for $j = 1, 2, \dots, J$ from the full conditional distribution

$$\pi(y_j^{(i)} | x^{(i)}, \xi_j^{(i)}) \propto \exp\{-Jq(\xi_j^{(i)})\} \mathbb{I}(Tx^{(i)} + Wy_j^{(i)} \geq h).$$

In so doing, generate a candidate $y_{j*}^{(i)} \sim g_3(y_{j*}^{(i)} | x^{(i)}, \xi_j^{(i)})$ and evaluate $u_3(x^{(i)}, \xi_j^{(i)}, y_{j*}^{(i)})$. Assuming g_3 is a symmetric proposal density, compute

$$a_3 = \min\left\{1, \frac{u_3(x^{(i)}, \xi_j^{(i)}, y_{j*}^{(i)})}{u_3(x^{(i)}, \xi_j^{(i)}, y_j^{(i)})}\right\}.$$

With probability a_3 , set $y_j^{(i)} = y_{j*}^{(i)}$ otherwise, keep $y_j^{(i)} = y_j^{(i-1)}$.

5. Set $i = i + 1$. Repeat steps 2 through 4 until the Markov chain is judged to have practically converged.

In what follows we illustrate an implementation of Algorithm 2. Consider the following simple two-stage stochastic program

$$\begin{aligned}
 & \max_x \quad 3x - E[Q(x, \xi)] \\
 & \text{subject to} \quad x \leq 3, x \geq 0 \\
 & \text{where } Q(x, \xi) = \min_y \quad \xi y \\
 & \text{subject to} \quad 0.5x + y \geq 5, y \geq 0,
 \end{aligned} \tag{12}$$

where ξ has a lognormal distribution with parameters $\mu = 0$ and $\sigma^2 = 0.1$. It is easy to see that the optimal solution for the first stage problem is $x^* = 3$. The optimal value for the decision variable is 3, no matter the number of scenarios sampled, whereas the objective function value can change with the realizations of the random variable. Our augmented probability distribution is given by

$$\pi_J(x, \boldsymbol{\xi}_J, \mathbf{y}_J) \propto \prod_{j=1}^J (3x - \xi_j y_j) p(\xi_j) \mathbb{I}(0 \leq x \leq 3, y_j \geq 5 - 0.5x, y_j \geq 0).$$

In implementing Algorithm 2, we use a Gibbs sampler where the conditional distributions are given by

$$\begin{aligned}
 \pi(x | \boldsymbol{\xi}_J, \mathbf{y}_J) & \propto \prod_{j=1}^J (3x - \xi_j y_j) p(\xi_j) \mathbb{I}(0 \leq x \leq 3) \\
 \pi(\xi_j | x, y_j) & \propto (3x - \xi_j y_j) p(\xi_j) \\
 \pi(y_j | x, \xi_j) & \propto \exp\{-J \xi_j y_j\} \mathbb{I}(y_j \geq \max(0, 5 - 0.5x)),
 \end{aligned}$$

for $j = 1, \dots, J$, where we use the Pincus (1968) result to obtain the conditional distribution for ξ_j . We use our algorithm and the Metropolis method with normal proposal distributions to draw from the above distributions.

We run 5000 iterations of our Gibbs sampler after an initial 500 burn-in samples. In Figure 11 we present the trace plots from the simulations of decision variable x for values $J = 2, 10, 20, 100$. The plots show that the algorithm converges to the true value of x as J increases. For $J = 20$, we estimate the true value of the first stage decision variable as 2.998, with a Monte Carlo error of 0.06%.

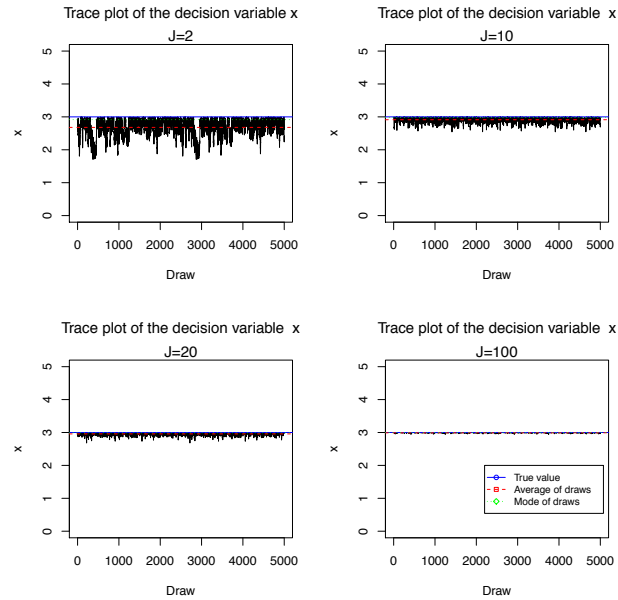


Figure 11 Trace plots for the decision variable x for different values of J

References

- Aarts, Emile, Jan Korst. 1988. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley and Sons, Inc., New York.
- Bayraksan, Guzin, David P Morton. 2009. Assessing solution quality in stochastic programs via sampling. *Tutorials in Operations Research* **5** 102–122.
- Benders, J.F. 1962. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* **4** 238–252.
- Bielza, Concha, Peter Müller, David Ríos Insua. 1999. Decision analysis by augmented probability simulation. *Management Sci.* **45**(7) 995–1007. doi:<http://dx.doi.org/10.1287/mnsc.45.7.995>.
- Birge, John R, Francois Louveaux. 2011. *Introduction to stochastic programming*. Springer Verlag, Berlin.
- Birge, J.R., F.V. Louveaux. 1988. A multicut algorithm for two stage stochastic linear programs. *Eur. J. of Oper. Res.* **34** 384–392.
- Brooks, Stephen P., Gareth O. Roberts. 1997. Assessing convergence of markov chain monte carlo algorithms. *Statistics and Computing* **8** 319–335.
- Dai, L, CH Chen, JR Birge. 2000. Convergence properties of two-stage stochastic programming. *Journal of Optimization Theory and Applications* **106**(3) 489–509.
- Dantzig, G.B. 1955. Linear programming under uncertainty. *Management Sci.* **1**(3-4) 197–206.
- Dantzig, G.B., M.N. Thapa. 1997. *Linear Programming Vol. 2. Theory and Extensions*. Springer-Verlag New York.

- Dempster, A.P., N.M. Laird, D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* **39** 138.
- Ermoliev, Y. 1988. Stochastic quasigradient methods. Y. Ermoliev, R.J-B. Wets, eds., *Numerical Techniques for Stochastic Optimization*. Springer Verlag, Berlin, 141–185.
- Gaivoronski, A. 1988. Stochastic quasigradient methods and their implementation. Y. Ermoliev, R.J-B. Wets, eds., *Numerical Techniques for Stochastic Optimization*. Springer Verlag, Berlin, 313–351.
- Gelman, Andrew, Donald B. Rubin. 1992. Inference from iterative simulation using multiple sequences. *Statistical Science* **7** 457–472.
- Geman, Donald, Stuart Alan Geman. 1987. Relaxation and annealing with constraints. Complex Systems Technical Report 35, Division of Applied Mathematics, Brown University.
- Higle, J.L., S. Sen. 1991. Stochastic decomposition: an algorithm for two stage linear programs with recourse. *Math. Oper. Res.* **16**(3) 650–669.
- Higle, J.L., S. Sen. 1996. *Stochastic decomposition: A statistical method for large scale stochastic linear programming*. Kluwer Academic Publishers.
- Homem-de Mello, Tito, Güzin Bayraksan. 2013. Monte carlo sampling-based methods for stochastic optimization. *Manuscript, under review for Surveys in Operations Research and Management Science*. Preprint available at *Optimization Online* (<http://www.optimization-online.org>).
- Infanger, Gerard. 1993. Monte carlo(importance) sampling within a benders' decomposition for stochastic linear programs. *Annals of Operations Research* **39** 69–95.
- Isaacson, Dean L, Richard W Madsen. 1976. *Markov chains, theory and applications*, vol. 4. Wiley New York.
- Jacquier, Eric, Michael Johannes, Nicholas Polson. 2006. Maximum expected utility via mcmc. Unpublished manuscript, Cirano HEC Montreal, Montreal, QC.
- Jacquier, Eric, Michael Johannes, Nicholas Polson. 2007. Mcmc maximum likelihood for latent state models. *Journal of Econometrics* **137** 615–640.
- Jonsbraten, Tore W., Roger J.-B. Wets, David L. Woodruff. 1998. A class of stochastic programs with decision dependent random elements. *Annals of Operations Research* **82** 83–106.
- Kim, Sujin, Raghu Pasupathy, Shane G Henderson. 2011. A guide to sample-average approximation. Unpublished manuscript, Virginia Tech University, Blacksburg, VA.
- Kirkpatrick, S., C. Gelatt, M. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.
- Mak, Wai-Kei, David P. Morton, R. Kevin Wood. 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24** 47–56.
- Mitra, Sovan. 2008. Scenario generation for stochastic programming. OptiRisk Systems: White Paper Series Domain: Finance Reference Number: OPT 004 URL:<http://www.optirisk-systems.com/papers/opt004.pdf>.

- Müller, Peter. 1999. Simulation-based optimal design. J.M. Bernardo, A.F.M. Smith Berger J.O., A.P. Dawid, eds., *Bayesian Statistics 6*. Oxford University Press, 459–474.
- Müller, Peter, Bruno Sansó, Maria De Iorio. 2004. Optimal bayesian design by inhomogeneous markov chain simulation. *Journal of the American Statistical Association* **99**(467) 788–798.
- Parpas, Panos, Berk Ustun, Mort Webster. 2012. Importance sampling in stochastic programming: A markov chain monte carlo approach. Working Paper Available at <http://web.mit.edu/mort/www/Parpasetal2012.pdf>.
- Pennanen, Teemu, Matti Koivu. 2005. Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische mathematik* **100**(1) 141–163.
- Pincus, Martin. 1968. A closed form solution of certain programming problems. *Oper. Res.* **16**(3) 690–694.
- Pincus, Martin. 1970. A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Oper. Res.* **18**(6) 1225–1228.
- Polson, N., M. Sorensen. 2011. A simulation-based approach to stochastic dynamic programming. *Applied Stochastic Models in Business and Industry* **27** 151–163.
- Rubinstein, Reuven Y, Alexander Shapiro. 1993. *Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method*, vol. 346. Wiley New York.
- Shapiro, Alexander. 2003. Monte carlo sampling methods. *Handbooks in operations research and management science* **10** 353–425.
- Shapiro, Alexander, Darinka Dentcheva, Andrzej P Ruszczyński. 2009. *Lectures on stochastic programming: modeling and theory*, vol. 9. SIAM.
- Shapiro, Alexander, Tito Homem-de Mello. 1998. A simulation-based approach to two-stage stochastic programming with recourse. *Math. Programming* **81**(3) 301–325.
- Shapiro, Alexander, Tito Homem-de Mello. 2000. On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs. *SIAM J. Optim.* **11**(1) 70–86.
- Spall, James C. 2005. *Introduction to stochastic search and optimization: estimation, simulation, and control*, vol. 65. John Wiley & Sons.
- Tierney, L. 1994. Markov chains for exploring posterior distributions (with discussion). *The Annals of Statistics* **22**(4).
- van Laarhoven, P.J., E.H.L. Aarts. 1987. *Simulated Annealing: Theory and Applications*. Reidel, Amsterdam.
- van Slyke, R.M., R. Wets. 1969. L-shaped linear programming with application to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17** 638–663.