Available online at www.sciencedirect.com
SCIENCE DIRECT
Computers and Chemical Engineering 30 (2006) 864–877
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

# An efficient algorithm for large scale stochastic nonlinear programming problems

Y. Shastri [a,b], U. Diwekar [a,*]

[a] *Vishwamitra Research Institute, Center for Uncertain Systems: Tools for Optimization and Management, 34 N. Cass Avenue, Westmont, IL 60559, USA*

[b] *University of Illinois at Chicago, Department of Bioengineering, 851 S. Morgan Street, Chicago, IL 60607, USA*

Received 22 October 2004; received in revised form 16 November 2005; accepted 27 December 2005
Available online 7 February 2006

**Abstract**

The class of stochastic nonlinear programming (SNLP) problems is important in optimization due to the presence of nonlinearity and uncertainty in many applications, including those in the field of process systems engineering. But despite the apparent importance of such problems, the solution algorithms for these problems have found few applications due to the severe computational and structural restrictions. To that effect, this work proposes a new algorithm for a computationally efficient solution of the SNLP problems. Starting with the basic structure of the traditional L-shaped method, the new algorithm, called the L-shaped BONUS, incorporates the reweighting scheme to ease the computational load in the second stage recourse function calculation. The reweighting idea has previously been successfully used in optimization in BONUS, also an algorithm to solve the SNLP problems. The proposed algorithm is analyzed using different case study problems, including a blending problem relevant to the process industry and a large scale, novel sensor placement problem for water security networks. The results for all the problems show considerable savings in the computational time without compromising the accuracy, the performance being better for the Hammersley sequence sampling technique as compared to the Monte Carlo sampling technique.


*PACS:* 89.60.+x

*Keywords:* Stochastic programming; Nonlinear programming; Water security network; BONUS

## 1. Introduction

The stochastic nonlinear programming (SNLP) problems represent an important class of the optimization problems due to their omnipresence in real life situations. Many systems in nature are inherently nonlinear, necessitating nonlinear models for their representation, and consequently, nonlinear programming methods for optimization. Another important factor for consideration is uncertainty. Very rarely are the system details accurately known. Quite often, the parameters and variables are known only in terms of their ranges or, in some cases, in terms of their probability distributions. In such cases, the stochastic programming methods need to be resorted to for optimization.

The field of process systems engineering is also replete with the applications of stochastic programming, many of which are nonlinear. Numerous well known tasks in this field, such as project planning and scheduling, chemical synthesis, process design and optimization, and some new fields, such as computer aided molecular design, use stochastic programming. An extensive review of the stochastic programming methods and their applications in the process engineering field is given in Diwekar (2003c) and Sahinidis (2004). Some of the recent applications include enterprise-wide process network (Rya, Dua, & Pistikopoulos, 2004), planning and scheduling related tasks (Jung, Blau, Penky, Reklaitis, & Eversdyk, 2004; Lin, Janak, & Floudas, 2004) and environment related applications (Diwekar, 2003a, 2005; Kheawhom & Hirao, 2004). Many of these problems are nonlinear, complicating the problem solution.

A general stochastic nonlinear programming problem can be represented as follows:

$$\begin{aligned} \text{Optimize} \quad J &= P_1(f(\theta, x, u)) \\ \text{such that} & \\ P_2(g_1(\theta, x, u)) &= 0 \\ P_3(g_2(\theta, x, u) \leq 0) &\geq \alpha \end{aligned}$$

* Corresponding author. Tel.: +1 630 5158772; fax: +1 312 9965921.
*E-mail address:* urmila@vri-custom.org (U. Diwekar).

0098-1354/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2005.12.009

where $\theta$ is a set of decision variable, $x$ is a set of system parameters, $u$ is a set of uncertain variables, and $P_1$, $P_2$ and $P_3$ are the probabilistic measures such as the expected value or variance. The uncertainty may affect the objective function and/or any of the constraints to make it a stochastic programming problem.

Even though important, solution of these SNLP problems is hard due to the inherent complexity and various limitations of the available solution algorithms, high computational requirements being one of them. This work proposes a new algorithm, the L-shaped BONUS, to solve the large scale stochastic nonlinear programming problems in a computationally efficient manner. The proposed algorithm is an integration of the traditional sampling based L-shaped method with a new algorithm BONUS (Better Optimization of Nonlinear Uncertain Systems), proposed to solve the SNLP problems. The L-shaped BONUS algorithm is shown to have better computational properties through an illustrative example, a process systems engineering relevant problem and a large scale optimization problem. The algorithm can also be used to convert an SNLP problem into an stochastic linear programming (SLP) problem.

The next section briefly overviews the common SNLP solution algorithms, and gives the motivation for proposing a new algorithm by looking at their limitations. Section 3 elaborates on the BONUS algorithm along with the reweighting scheme, which is central to the new algorithm, and Section 4 reviews the sampling based L-shaped algorithm. Section 5 explains the integration of these concepts into the proposed L-shaped BONUS algorithm. Sections 6–8 give the details of the algorithm steps and the computational advantages through various case study problems. The final section draws the concluding remarks.

## 2. Methods for SNLP problem solution

### 2.1. General overview

Over the years, a lot of research has gone in devising the strategies to solve the SNLP problems. One kind of the solution methods, such as the chance constraint programming method (Charnes & Cooper, 1959), convert these problems into the deterministic equivalents. The deterministic optimization methods can then be applied. But these methods are restricted to the problems with known and stable density functions of the random variables. The second kind of solution methods are aimed at extending the deterministic nonlinear programming methods to included uncertainty (Elmaghraby, 1960; Ziemba, 1970, 1972). For the optimization problems that can be decomposed into two or multiple stages, decomposition based stochastic programming methods, such as the L-shaped method, are developed (Birge & Louveaux, 1997). The basic L-shaped method is modified for the various problem types, which includes the regularized decomposition method (Ruszczyński, 1986) and the piecewise quadratic form of the L-shaped method (Louveaux, 1978). The methods based on the stochastic programming Lagrangian include the basic Lagrangian dual ascent method (Birge & Louveaux, 1997), the Lagrangian finite generation method for linear quadratic stochastic programs (Rockafellar & Wets, 1986) and the progressive hedging algorithm (Rockafellar & Wets, 1991). These decomposition based methods require convexity of the problem and/or a dual block angular structure. The stochastic quasi-gradient methods (SQG) are less specialized than the other algorithms, but are useful to solve the problems having complex objective functions and constraints (Louveaux, 2001). The SQG methods represent one of the first computational developments in stochastic programming. An exhaustive review of all these methods is omitted for brevity, and the reader is referred to the respective references made for these solution algorithms. Even though important, application of these methods to solve the real life problems has always been restricted. This is due to the various limitations in the form of functional requirements (convexity, differentiability, etc.) or distribution of uncertain variables (stable).

### 2.2. Sampling based methods

In stochastic programming problems, it is common to use sampling approximations when the probability distributions of the uncertain parameters are known. The aim is to model the complete uncertain parameter space as closely as possible through a sufficient number of samples. Fig. 1 represents the generalized solution procedure for the sampling based approach. The structure is similar to that for a deterministic problem, apart from the fact that the deterministic model is replaced by a stochastic model with the (shaded) sampling loop representing the discretized uncertainty space. The goal in stochastic programming is to improve the probabilistic objective function value with each iteration. In the set up of Fig. 1, the calculation of these terms requires the simulation of the stochastic modeler at each iteration. In the traditional sampling based methods, this is achieved by model simulations for a given number of samples, and the subsequent computation of the probabilistic function (e.g. expected value of the objective function). Two such methods are the sampling based L-shaped method (Dantzig & Glynn, 1990; Dantzig & Infanger, 1992) and the Stochastic Decomposition algorithm (Higle & Sen, 1991). The L-shaped method is a scenario based method. But the number of scenarios increases

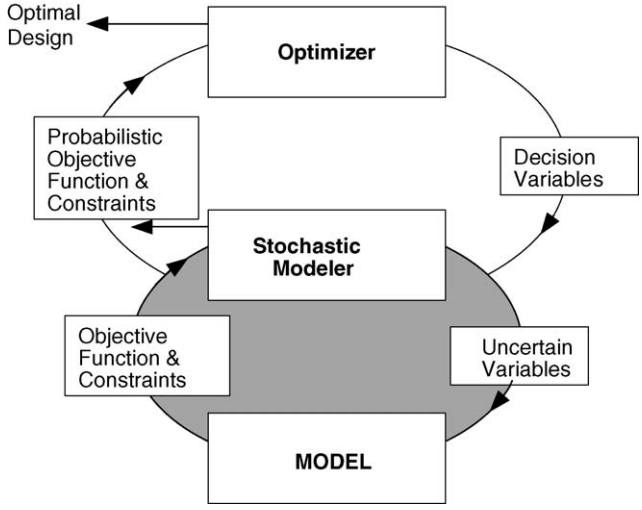


Fig. 1. Optimization under uncertainty.

exponentially as the number of uncertain variables increases. Monte Carlo sampling avoids this problem, and hence, Monte Carlo sampling based approximations have been incorporated in the L-shaped method. The key feature is the use of statistical estimates to obtain confidence intervals on the results.

The simulation of the stochastic modeler at each iteration is a major drawback of the sampling based methods. For a sample size of $n$, the model needs to be simulated $n$ times in each iteration as a part of the stochastic modeler. With a larger sample size, required for better approximation, the computational load increases tremendously. The next section explains BONUS, an algorithm that overcomes this problem through the use of a reweighting scheme.

## 3. BONUS

The sampling based approaches to solve the stochastic nonlinear programming problems suffer from the main drawback of computational complexity, as mentioned in the previous section. Recently, a new method has been proposed to solve the SNLP problems that holds its advantage in circumventing the problem of repeated model simulations. This new method is Better Optimization of Nonlinear Uncertain Systems (BONUS) (Sahin & Diwekar, 2004). Fig. 2 shows the BONUS algorithm structure.

Fig. 1 shows the standard stochastic programming algorithm structure with the sampling loop requiring repeated model simulations. Compared to this, the BONUS algorithm uses a reweighting approach to skip these repeated model simulations. This reweighting scheme is central to the BONUS algorithm. An initial, uniform base distribution of the uncertain parameters is generated. For the first iteration, the algorithm emulates the standard sampling based algorithm in that the model is simulated for each sample to determine the output distribution. At the subsequent iterations, when the optimizer needs new estimates of the probabilistic values of the objective function, a new set of samples is taken. But this time the model is not re-run. Instead, the reweighting approach is applied to approximate the probabilistic behavior of the new output distribution. Fig. 2 illustrates this step of the algorithm. The reweighting scheme uses the initial sample set, the initial output distribution and the new sample set data to estimate the information about the new output distribution. Owing to the importance of reweighting in the proposed algorithm, it is explained in the following section.

### 3.1. The reweighting approach

The reweighting approach is based on the various reweighting schemes proposed by Hesterberg (1995). It is an extension of the importance sampling concept of estimating something about a distribution (target distribution $f(x)$) using observations from a different distribution (design distribution $\hat{f}(x)$), where these distributions are represented by the respective probability density functions. Let $X$ be a random variable with the probability density function $f(x)$, and $Q(X)$ be a function of $X$. Then, to estimate a certain property of $Q(X)$, such as the expected value $\mu = E_f[Q(X)]$, importance sampling solves a different

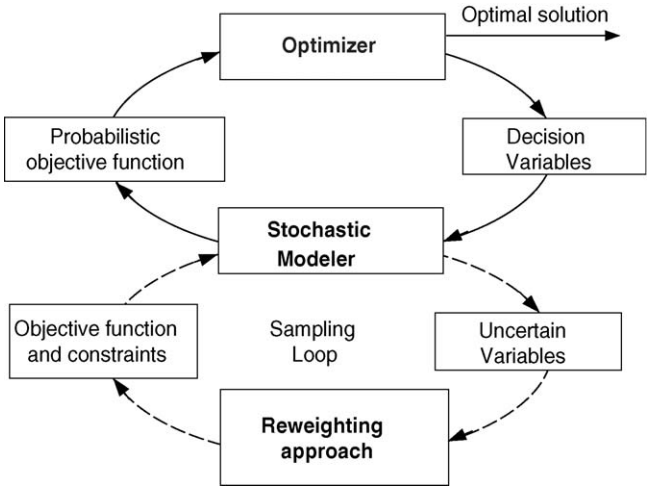


Fig. 2. BONUS algorithm.

problem of estimating $E_{\hat{f}}[Y(X)]$, where

$$Y(x) = Q(x)\frac{f(x)}{\hat{f}(x)} \tag{1}$$

and the samples $X_i$ are now drawn from $\hat{f}(x)$. The distribution $\hat{f}(x)$ can be designed to achieve the desired results (e.g. reduced variance, better representation of the rare events). The weight function $W(x)$ is defined as

$$W(x) = \frac{f(x)}{\hat{f}(x)} \tag{2}$$

which gives the likelihood ratio between the target and the design distributions and weighs observations of $Q(x)$. To perform this estimation effectively, Hesterberg (1995) proposed various design distributions $\hat{f}(x)$ (e.g. defensive mixture distributions) and estimation schemes (integration estimate, ratio estimate). In the ratio estimate, the weights $W_i$ are normalized to avoid problems when they do not sum to 1. The normalized weights $V_i$ and the estimate $\mu$ is given as

$$V_i = \frac{W_i}{\sum_{j=1}^{n} W_j} \tag{3}$$

$$\mu = \sum_{i=1}^{n} V_i Q(X_i) \tag{4}$$

where $n$ is the sample size. Means, higher moments and percentiles can be computed using such relations. The reweighting scheme in the proposed algorithm is based on the ratio estimate just explained.

The reweighting approach, as used in the BONUS algorithm, is schematically shown in Fig. 3. Suppose $X$ represents the uncertain variable in the stochastic programming problem, and $Q(X)$ is the output of the stochastic modeler. For the first iteration, the base case samples $X_i^*$ with a uniform distribution ($\hat{f}(x)$) are drawn and the model is simulated for each sample to get the complete model output distribution $Q(X_i^*)$. During the subsequent iterations, new samples $X_i$ of the required distribution ($f(x)$) are drawn. Having known the model response $Q(X_i^*)$ for

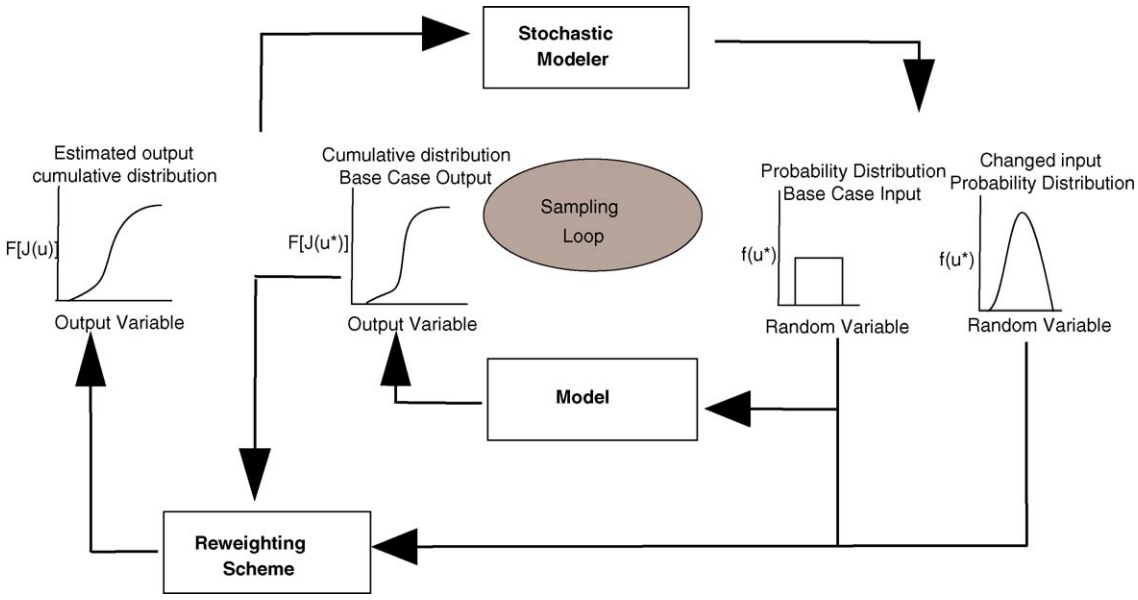


Fig. 3. Reweighting approach in the BONUS algorithm.

the sample set $X_i^*$ from the distribution $\hat{f}(x)$, it is possible to use Eq. (4) to estimate the expected value of the model response $Q(X_i)$ for the new sample set $X_i$ from the distribution $f(x)$. The expected value of the stochastic model response $Q(X_i)$ for the new sample set $X_i$ is therefore given as

$$E_f[Q(X_i)] = \sum_{j=1}^{n} \frac{(f(X_j)/\hat{f}(X_j^*))}{\sum_{i=1}^{n}(f(X_i)/\hat{f}(X_i^*))} Q(X_j^*) \tag{5}$$

In a sampling based algorithm, this procedure requires determining the probability density function from the available sample set. This is carried out using the Gaussian Kernel Density Estimation technique (Silvermann, 1986), which is a nonparametric density estimation technique. The basic idea behind this technique is to place a bin of a certain width $2h$ around every sample $X$ and weigh that sample by the number of other samples $X_i$ in the same bin. If this bin is replaced by a kernel function, such as the normal density function, the density function for the sample set $X_i$ is calculated using Eq. (6):

$$f(X) = \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-(1/2)((X-X_i)/h)^2} \tag{6}$$

where $h$ is the window width, also called the smoothing parameter or bandwidth. The value of $h$ decides the fineness of the density estimation. For this work, it is taken as the standard deviation of the sample set.

Thus, given the two sample sets, Eq. (6) is used to determine the density function at each sample point for both the distributions, which are then used in Eq. (5) to find out the output distribution for the second sample set.

For the proposed algorithm, this idea of reweighting is used in the sampling based L-shaped method to solve the decomposable SNLP problems in a computationally efficient manner. Before describing the proposed algorithm however, it is prudent to understand the L-shaped method, which is explained in the next section.

## 4. L-shaped method with sampling

The basic L-shaped method is a scenario based method applicable for discrete distributions to solve two or multi stage stochastic programming problems (Van Slyke & Wets, 1969). The basic idea of the L-shaped method is to approximate the nonlinear term in the objective function of the problem. The principle behind this approach is that, since the nonlinear objective function term (recourse function) involves the solution of all the second stage recourse problems, numerous function evaluations for it are avoided. This term therefore is used to build a master problem (with first stage decision variables) and the recourse function is exactly evaluated only as a subproblem, referred to as the second stage problem. Fig. 4 shows the algorithm structure.

The method is essentially a Dantzig–Wolfe decomposition (inner linearization) of the dual or a Benders decomposition (outer linearization) of the primal. The problem is decomposed into two or multiple stages. The first stage problem (master problem) uses a linear approximation (also the lower bound) of the second stage recourse function to fix the first stage decision variables. These first stage decisions are passed on to the second stage (sub problem), where the dual of the second stage problem is solved for different scenarios. The solution of all the dual problems is used to calculate the expected value of the recourse function, which is also its upper bound. Two kinds of cuts are sequentially generated, the feasibility cut and the optimality cut. These cuts are added to the master problem for a better approximation of the recourse function in the subsequent iterations. The algorithm terminates when the upper bound from the sub problem is less than or equal to the lower bound from the master problem (Diwekar, 2003b).

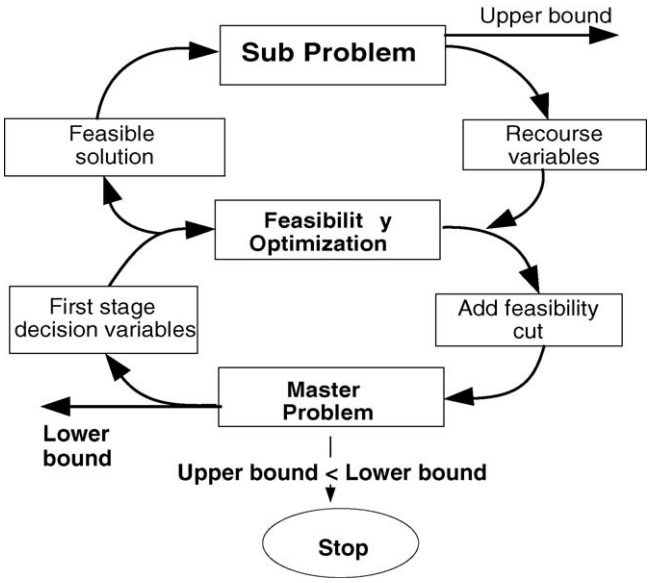


Fig. 4. L-shaped algorithm structure.

For continuous distributions, sampling is used to approximate the distribution. The use of importance sampling as a variance reduction technique was proposed by Dantzig and Glynn (1990), while the use of sampling in the basic L-shaped method was proposed by Dantzig and Infanger (1992). Samples, instead of scenarios, are used in the second stage recourse function calculations of the L-shaped method. Statistical approximations of the recourse function and the simplex multipliers are used to generate the cuts and the bounds. This method is known as the internal sampling method. In the context of the sampling based algorithms explained in Section 2, the sub-problem solution for each realization of the uncertain space can be compared with the stochastic modeler in Fig. 1.

Thus to summarize, the decomposition strategy of the L-shaped method offers computational savings, but is still not very efficient for the SNLP problems. Therefore, we are proposing an integration of the decomposition structure and the reweighting scheme. In the next section, the proposed L-shaped BONUS algorithm is detailed which achieves the same.

## 5. Proposed algorithm: L-shaped BONUS

The sampling based algorithm suffers from the computational bottleneck of repeated model simulations, while BONUS uses the reweighting approach to bypass this problem. The proposed algorithm is an integration of the sampling based L-shaped method with BONUS. The central idea of reweighting in BONUS is utilized in this algorithm. The modification is in the second stage recourse function calculation procedure of the L-shaped method. Since the structure of the algorithm is based on the L-shaped method and the application of the reweighting concept is similar to that in BONUS, the mathematical details are not reproduced here, and those can be found in Sections 3 and 4. The proposed algorithm is shown schematically in Fig. 5 and explained below.

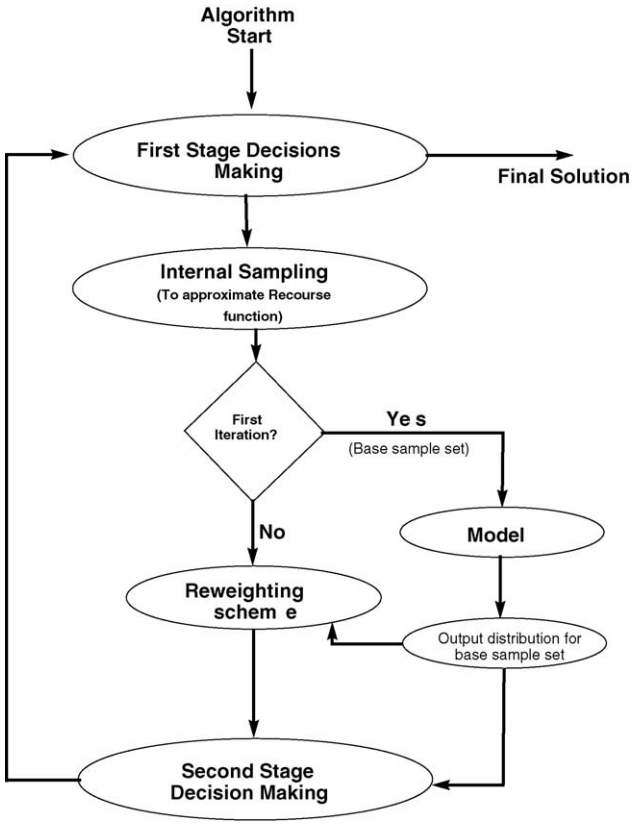


Fig. 5. Proposed L-shaped BONUS algorithm structure.

### 5.1. Algorithm details

The given stochastic programming problem is first converted into a two stage stochastic programming problem with recourse. The first stage decisions are taken using a linearized approximation of the second stage nonlinear recourse function and utilizing the feasibility and the optimality cuts, if generated. This also determines the lower bound for the objective function value. The second stage objective is the expected value of the recourse function, which depends on the first as well as the second stage decision (recourse) variables. Following the sampling based L-shaped method structure, the first stage decisions are passed on to the second stage, where the sub-problem is solved for each uncertainty realization. The idea of the proposed algorithm is to reduce the computations at the sub-problem solution stage by using the reweighting scheme to bypass the nonlinear model computations. The reweighting scheme, as mentioned in Section 3, needs the model output distribution for a base case uniform input distribution. For this purpose, during the first optimization iteration, the nonlinear model is simulated, and the sub-problem is solved for each sample. The model simulation results for the base case constitute the base case output distribution. The sub-problem solution for each sample is used to derive the optimality cut for the master problem and generate an upper bound for the objective function value according to the L-shaped algorithm. The second optimization iteration solves the first stage master problem using these cuts. The new first stage decisions along with an updated lower bound are

passed on to the second stage problem. During this iteration, when the new set of samples is taken by the stochastic modeler, the model simulation and the sub-problem solution steps are not performed for each sample. The reweighting scheme, with Gaussian Kernel Density Estimation, as explained in Section 3, is used to predict the probabilistic values (expectation) of the model output. The base case output distribution along with the two sample sets are used for this prediction. The expected value of the model output is used to solve only one second stage dual sub-problem to generate the cuts and to update the upper bound on the objective function value. It should be noted that for the second iteration, only one sub-problem is solved. Thus, not only the nonlinear model simulation time but also the sub-problem solution time is saved. This procedure of reweighting based estimation is continued in every subsequent iteration till the termination criteria based on the L-shaped method is encountered.

The primary advantage of the proposed L-shaped BONUS algorithm, as has been repeatedly stressed, is its computational efficiency. Repeated model simulations, which are a bottleneck in the stochastic optimization procedure, being avoided, the problem solution becomes faster. The effect is expected to be more pronounced in the case of nonlinear and/or high dimensional models, as the ones often encountered in the real life applications. Another advantage of this algorithm is its ability to convert an SNLP problem into an SLP problem by using reweighting to approximate the nonlinear relationships (see Section 8 for such an application).

The disadvantage is that reweighting is an approximation and the quality of this approximation is a point of contention. It has been shown that the estimation accuracy of reweighting improves with the increasing sample size, which also increases the computational load to a certain extent. The exact quantitative nature of this relationship is difficult to establish. It is thought that it will depend on the particular nonlinear system. For the details, the readers are referred to Sahin and Diwekar (2004).

Finally, as with any sampling based optimization technique, the sampling properties are very important for this algorithm. The accuracy of the reweighting scheme depends on the number and uniformity of the samples (see Section 8). For this algorithm, we propose to use the Hammersley Sequence Sampling (HSS), which is shown to be very efficient (Kalagnanam & Diwekar, 1997; Wang, Diwekar, & Grégorie Padró, 2004). The sampling technique is based on the generation and inversion of the Hammersley points, and is shown to have $k$ dimensional uniformity property.

## 6. Illustrative example: farmer's problem

This section explains the application of the proposed algorithm through a simple illustrative farmer's problem that has been extensively studied in the field of stochastic programming (Birge & Louveaux, 1997). The problem, as formulated in Birge and Louveaux (1997), is a stochastic linear programming problem, which is modified into an SNLP problem.

Table 1
Data for the farmer's problem

| | Wheat | Corn | Sugar beets |
|---|---|---|---|
| Yield (t/acre) | 2.5 | 3.0 | 20 |
| Planting cost (US$/acre) | 150 | 230 | 260 |
| Selling price (US$/t) | 170 | 150 | 36 under 6000 t<br>10 above 6000 t |
| Purchase price (US$/t) | 238 | 210 | – |
| Minimum requirement (t) | 200 | 210 | – |

### 6.1. Problem formulation

The goal of the problem is to decide the optimal allocation of 500 acres of plantation land amongst three crops: wheat, corn and sugar. The farmer needs at least 200 tonnes (t) of wheat and 240 t of corn for cattle feed. These amounts can be produced on the farm or bought from a wholesaler. The excess production can be sold in the market. The purchase cost is 40% more than the selling cost due to the wholesaler's margin and transportation cost. Sugar beet sells at a cost of US$ 36/t if the amount is less than 6000 t. Any additional quantity can be sold at US$ 10/t only. Through experience, the farmer knows that the mean yield of the crops is 2.5, 3 and 20 t per acre for wheat, corn and sugar, respectively. But these values are uncertain owing to various factors. The objective is to maximize the expected profit in the presence of the uncertain yields. Table 1 summarizes the data and more details about the SLP can be found in Birge and Louveaux (1997).

For this illustration, to convert the problem into an SNLP problem, the uncertain yield is assumed to be dependent on four different uncertain factors. These four factors are: average rainfall, availability of sunlight, attack probability of a crop disease and probability of attack by pests. The annual yield of the crops is *nonlinearly* related to these four factors. Although the relationships presented here are hypothetical and simplistic, it is expected that some nonlinear equations will govern these relationships. The dependencies are as follows:

$$Y_r = 2\alpha_r\left(1 - \frac{\alpha_r}{2}\right), \quad \alpha_r \in [0, 2] \tag{7}$$

$$Y_s = 1.58(1 - e^{-\alpha_s}), \quad \alpha_s \in [0, 1] \tag{8}$$

$$Y_d = 1 - \alpha_d, \quad \alpha_d \in [0, 1] \tag{9}$$

$$Y_p = 1 - \alpha_p^2, \quad \alpha_p \in [0, 1] \tag{10}$$

where $Y_i$ is fractions of the maximum yield due to the corresponding effects, $\alpha_r$ fractional rainfall of the yearly average, $\alpha_s$ fractional sunlight of the yearly average, $\alpha_d$ attack probability of a crop disease and $\alpha_p$ is attack probability of pests.

The overall fractional yield of the crops is given by

$$Y_{actual} = Y_r \times Y_s \times Y_d \times Y_p \times Y_{max} \tag{11}$$

where, $Y_{actual}$ is the actual yield of the crops and $Y_{max}$ is the maximum possible yield, if all the conditions are perfect. Once these equations are incorporated in the original model, the resulting

stochastic programming problem is given as:

$$\text{Minimize} \quad 150x_1 + 230x_2 + 260x_3 + E[238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4]$$

subject to the following constraints

$$x_1 + x_2 + x_3 \leq 500,$$
$$t_1(\xi)x_1 + y_1 - w_1 \geq 200,$$
$$t_2(\xi)x_2 + y_2 - w_2 \geq 240,$$
$$w_3 + w_4 \leq t_3(\xi)x_3,$$
$$w_3 \leq 6000,$$
$$x_1, x_2, x_3, y_1, y_2, w_1, w_2, w_3, w_4 \geq 0$$

where $E$ is the expectation operator over the uncertain variables $\xi$. $t_i(\xi)$ is the yield of the crop $i$ given by Eq. (11) and *nonlinearly* related to the uncertain variables through Eqs. (7)–(10).

This problem, when converted into a two stage stochastic programming problem with recourse, is given as:

First stage problem:

$$\text{Min} \quad 150x_1 + 230x_2 + 260x_3 + \theta$$
$$\text{s.t.} \quad x_1 + x_2 + x_3 \leq 500,$$
$$G_l x + \theta \geq g_l \qquad l = 1 \ldots s,$$
$$x_1, x_2, x_3 \geq 0,$$

where $\theta$ is the linear approximation of the expected value of the recourse function. $x_1$, $x_2$ and $x_3$ constitute the first stage decision variables. The constraints include the problem defined constraints on the first stage decision variables and the optimality cuts applied during the iterations of the L-shaped method.

Second stage problem:

$$Q(x, \xi) = \min\{238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4\}$$
$$\text{s.t.} \quad t_1(\xi)x_1 + y_1 - w_1 \geq 200,$$
$$t_2(\xi)x_2 + y_2 - w_2 \geq 240,$$
$$w_3 + w_4 \leq t_3(\xi)x_3,$$
$$w_3 \leq 6000,$$
$$y_1, y_2, w_1, w_2, w_3, w_4 \geq 0,$$

Here, $y_1$, $y_2$, $w_1$, $w_2$, $w_3$ and $w_4$ are the second stage decision variables (recourse variables). The constraints on the recourse variables in the original problem are considered in the second stage problem solution.

### 6.2. Problem solution

The problem, when solved using the sampling based L-shaped method, involves the dual formulation of the nonlinear second stage problem and the solution of the dual problem in the second stage, for each sample from the given sample set. Even if the nonlinearity is separated from the problem by considering directly the yield in the second stage problem (in place of the nonlinear relationships), the task of the dual problem solutions for the samples can be demanding.

The proposed algorithm can simplify this task by using reweighting to bypass the nonlinear model, as represented by Fig. 5. The ability of reweighting to effectively model the nonlinear relationship between the uncertain parameters and the crop yield helps to convert the problem into an SLP problem with reduced computations.

The exact solution procedure is as follows. At every second stage problem solution, the uncertain parameters are sampled $n$ times, $n$ being a pre-decided sample size. During the first iteration, the samples are used to calculate the value of the crop yield, and the yield value is used to solve the dual for each sample (i.e. $n$ dual problem solutions), and the optimality cut, if needed, is generated. The first sample set is stored as the base sample set.

For the subsequent iterations, during the second stage problem solution, the new set of $n$ samples are taken. However, instead of solving the dual for each sample through yield calculation, reweighting is used to calculate the expected value of the crop yield. This single expected value is used in the dual problem, which is now converted into a linear one. Moreover, with one expected value of the yield, the dual problem needs to be solved only once to calculate the expected value of the recourse function and generate the cut if needed. The use of reweighting, therefore, simplifies the problem on two counts. First, it bypasses the nonlinear part of the model and converts it into a linear model, and then, the computations are simplified by solving just one problem at the second stage. Reproduced below are the the first two iterations of the problem solution to explain the steps.

**Solution: Iteration 1**

- Step 0: $s = 0$ (iteration count)
- Step 1: $\theta^1 = -\infty$ (very low value). Solve

$$\text{Min} \quad 150x_1 + 230x_2 + 260x_3$$
$$\text{s.t.} \quad x_1 + x_2 + x_3 \leq 500$$
$$x_1, x_2, x_3 \geq 0$$

The solution is: $x_1^1 = x_2^1 = x_3^1 = 0$.

- Step 2: Sample the uncertain variables $n$ times to generate the base sample set $\{u^*\}$
- Step 3: Calculate the yield ($n$ values) of the crops using $n$ sampled uncertain variables and relations (7)–(10).
- Step 4: Solve the following dual problem for the $n$ values of the crop yield. The values of $x_i^1$ are passed on to the second stage.

$$\text{Max} \quad \pi_1(200 - Y_1x_1^1) + \pi_2(240 - Y_2x_2^1) - \pi_3(Y_3x_3^1) - 6000\pi_4$$
$$\text{s.t.} \quad \pi_1 \leq 238$$
$$\pi_2 \leq 210$$
$$\pi_1 \geq 170 \qquad (12)$$
$$\pi_2 \geq 150$$
$$\pi_3 + \pi_4 \geq 36$$
$$\pi_3 \geq 10$$
$$\pi_1, \pi_2, \pi_3, \pi_4 \geq 0$$

The solution of problem (12) for the first sample is: $\pi_1 = 236$, $\pi_2 = 210$, $\pi_3 = 36$, $\pi_4 = 0$. The expected value of the recourse function ($w$), calculated after all the dual problem solutions, is $w = 98000$. Since $w > \theta$, an optimality cut is introduced.

**Iteration 2:**

- Step 0: $s = 1$ (iteration count)
- Step 1: Solve

$$\begin{aligned} &\text{Min} \quad 150x_1 + 230x_2 + 260x_3 + \theta \\ &\text{s.t.} \quad x_1 + x_2 + x_3 \le 500 \\ &\theta \ge 98000 - [\,610.1\ 636.4\ 727.4\,][\,x_1\ x_2\ x_3\,]^{\mathrm{T}} \\ &x_1, x_2, x_3 \ge 0 \end{aligned} \tag{13}$$

The solution of problem (13) is: $x_1 = 0$, $x_2 = 0$, $x_3 = 500$ and $\theta = -264685.247$.

- Step 2: Sample the uncertain variables $n$ times to generate the new sample set $\{u\}$
- Step 3: Calculate the estimated yield of the crops using the base and the new sample sets, bypassing relations (7)–(10). The estimated yield is 0.842.
- Solve the dual problem given by the equation set (12) only once, using the estimated value of the crop yield. The solution of the problem is: $\pi_1 = 238$, $\pi_2 = 210$, $\pi_3 = 10$, $\pi_4 = 26$. The expected value of the recourse function ($w$), calculated after the dual problem solution, is $w = -158986.526$. Another optimality cut is introduced.

The procedure is then followed according to iteration 2 (using reweighting instead of $n$ dual problem solutions) till the termination criteria of $w \le \theta$ is satisfied.

### 6.3. Results of the farmer's problem

The comparative results for the farmer's problem using the sampling based L-shaped method and the L-shaped BONUS algorithm are shown graphically in Figs. 6 and 7, which also compare the results for two different sampling techniques, Monte Carlo sampling (MCS) and Hammersley sequence sampling Fig. 6 compares the objective function values at the final solution as a function of the sample size. It is seen that the solutions for both the algorithms approach a steady state value with increasing sample size. Moreover, the difference in the results for the two algorithms is within reasonable limits, 1.7% for the maximum sample size, indicating that the reweighting based approximation is not sacrificing the accuracy. Fig. 7 gives the plots for one decision variable value (land allocation to crop 3). Qualitatively, it shows a similar variation as that for the objective function value.

Based on these plots, HSS emerges as the more efficient sampling technique than MCS. As the sample size is increased, the results for HSS appear to reach a steady state value faster than those for MCS. The claim is further corroborated by Fig. 8, which plots the iterations needed to reach the solution for the

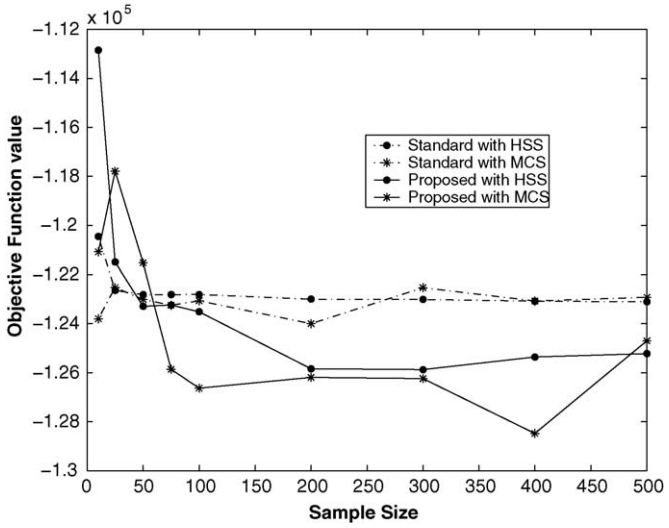


Fig. 6. Variation of the objective function value with sample size for the farmer's problem.

different sample sizes. It can be observed that for the standard L-shaped method, the MCS technique needs more iterations in general than the HSS technique. The previously mentioned $k$ dimensional uniformity property of HSS accounts for this observation. It has been previously shown that the number of points required to converge to the mean and the variance of a derived distributions by the HSS technique is on average 3– 100 times less than the MCS and other stratified sampling techniques (Kalagnanam & Diwekar, 1997). For the same sample size, the HSS method therefore approximates a given distribution better than the MCS method. This results in a faster convergence of the HSS based algorithms in general. For the proposed algorithm though, both, the MCS and the HSS techniques, need six iterations, irrespective of the sample size. This is possibly due to the approximation introduced by the reweighting scheme. The approximation renders the iteration requirements insensitive to the sample size and the sampling method changes. But the better values of the final

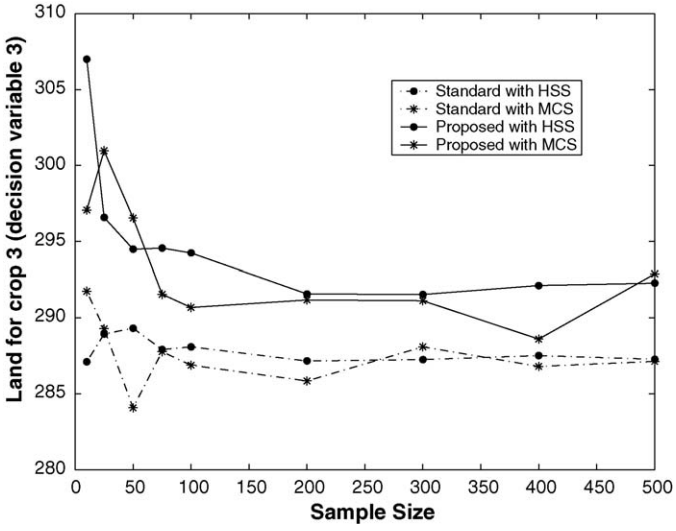


Fig. 7. Variation of the decision variable 3 value with sample size for the farmer's problem.

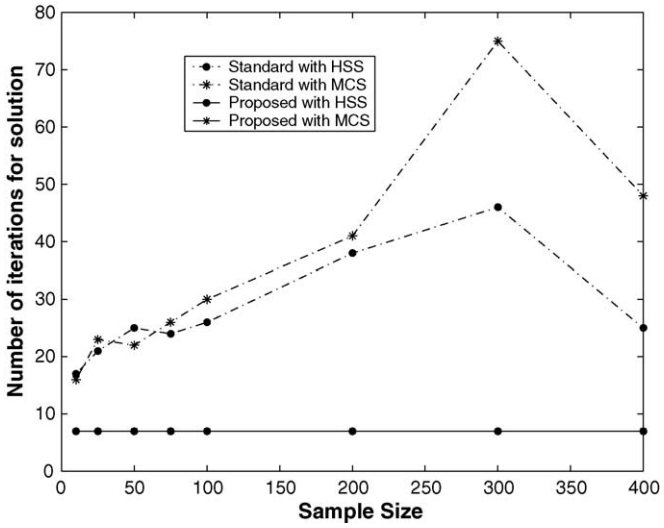


Fig. 8. Variation of number of iterations with sample size for the farmer's problem.

solutions (Figs. 6 and 7) confirm the superiority of HSS over MCS.

The computational time is an important factor while comparing these algorithms. The computational time increases exponentially with the sample size for the standard L-shaped method, while it increases almost linearly for the proposed L-shaped BONUS algorithm. The computational efficiency of the L-shaped BONUS therefore becomes more pronounced as the sample size is increased. With the need to increase the sample size to improve the accuracy, the proposed algorithm offers a distinct advantage.

The next section shows an application of the proposed algorithm to a process systems engineering relevant problem.

## 7. Blending problem

The problem reported here is typical for a petroleum industry manufacturing finished petroleum products such as lube oils. A large number of natural lubricating and specialty oils are produced by blending a small number of lubricating oil base stocks and additives. The lube oil base stocks are prepared from the crude oil by distillation. The additives in the form of chemicals are used to impart desirable characteristics and improve the existing properties of the base stocks (Gray, 1986; Rhodes, 1993). In the context of such an application, a general chemical blending optimization problem is explained below, followed by the results comparing different solution and sampling techniques.

### 7.1. Problem formulation

The aim is to blend $n$ different chemicals (such as lube oil base stocks and additives) to form $p$ different blend products (lube oils) at the minimum overall cost. Each chemical (base stock) has varying fractions of $m$ different components (such as $C_1$–$C_4$ fraction, $C_5$–$C_8$ fraction, heavy fraction, inerts, etc.). The market demands call for the production of a particular quantity of each blend product. The blend products catering to different applications (e.g. high performance lube oil, grease, industrial grade lube oil, etc.) have different specifications on the fractions of $m$ different components (for a lube oil such specifications will depend on the physical property requirements like pour point, viscosity, boiling temperature). These specifications need to be satisfied to sell the blend products. The task is complicated due to the presence of $q$ impurities in the chemicals. The exact weight fractions of these impurities in some of the chemicals (base stocks) are uncertain. Such uncertainties may arise when the chemicals to be blended are themselves a product of other processes (such as crude distillation for lube oil base stocks). There also are specifications on the maximum amount of impurity in a blend product. If the impurity content of a blend product does not satisfy the regulation, the product has to be treated to reduce the impurities below specifications. The treatment cost depends on the amount of reduction in the impurities to be achieved. The goal in formulating a stochastic optimization problem is to find the optimal blend policy to minimize the raw material cost and the expected blend product treatment cost in the presence of uncertainty associated with the impurity content of the chemicals. The stochastic programming problem is formulated as follows:

$$\text{Minimize} \qquad \sum_{i=1}^{n}\sum_{k=1}^{p} C_i W_{ik} + E\left[\sum_{k=1}^{p} C_T \theta_k\right] \tag{14}$$

Subject to:

$$\sum_{i=1}^{n} W_{ik} = \bar{W}_k, \quad \forall k = 1, \ldots, p \tag{15}$$

$$\sum_{i=1}^{n} x_{ij} W_{ik} \geq \bar{x}_{jk}, \quad \forall k = 1, \ldots, p \text{ and } j = 1, \ldots, m \tag{16}$$

$$\sum_{l=1}^{q} (I_{il}(u))^{\alpha_l} = I_i^*, \quad \forall i = 1, \ldots, n \tag{17}$$

$$\sum_{i=1}^{n} I_i^* W_{ik} = \bar{I}_k, \quad \forall k = 1, \ldots, p \tag{18}$$

$$\bar{I}_k.(1 - \theta_k) \leq I_k^{\text{spec}}, \quad \forall k = 1, \ldots, p \tag{19}$$

Here, $W_{ik}$ is the total weight of chemical $i$ in blend product $k$. $C_i$ is the per unit cost of chemical $i$, while $C_T$ is the blend product treatment cost per unit reduction in the impurity content. $\bar{W}_k$ is the total production requirement (in weight units) of blend product $k$. $x_{ij}$ is the weight fraction of component $j$ in chemical $i$, and $\bar{x}_{jk}$ is the specification (in weight units) of component $j$ in blend product $k$. $I_{il}(u)$ is the (possibly uncertain) weight fraction of impurity $l$ in chemical $i$, and $I_i^*$ is the 'impurity parameter' of chemical $i$. This impurity parameter gives the extent to which a chemical is impure, as a nonlinear function of the various impurities. The coefficients $\alpha_l$ decide the importance of a particular impurity in the final product. $\bar{I}_k$ is the final impurity parameter of a blend, which depends on the weight contribution of each chemical in a particular blend. $I_k^{\text{spec}}$ is the maximum permitted impurity content in the blend product. $\theta_k$ is the purification required for blend $k$ to satisfy the impurity constraint.

Table 2
Data for the chemicals in the blending problem

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ |
|---|---|---|---|---|---|---|---|
| $C_1$ fraction | 0.20 | 0.10 | 0.50 | 0.75 | 0.10 | 0.30 | 0.20 |
| $C_2$ fraction | 0.10 | 0.15 | 0.20 | 0.05 | 0.70 | 0.30 | 0.55 |
| $C_3$ fraction | 0.60 | 0.65 | 0.22 | 0.12 | 0.10 | 0.30 | 0.16 |
| $I_1$ fraction | 0.02 | 0.07 | 0.01 | 0.02 | 0.043 | 0.015 | 0.012 |
| $I_2$ fraction | 0.01 | 0.005 | 0.02 | 0.02 | 0.01 | 0.04 | 0.021 |
| $I_3$ fraction | 0.06 | 0.023 | 0.02 | 0.03 | 0.022 | 0.028 | 0.055 |
| Cost (US$/unit weight) | 104 | 90 | 135 | 130 | 115 | 126 | 120 |

The objective function consists of two parts. The first part is the cost of the chemicals used to manufacture the blend products, and the second part is the expected treatment cost of the off-spec products. The first set of constraints ensures the required production of each blend product. The second constraint set ensures that the component specifications for the blended products are satisfied. These specifications are expressed in terms of the minimum amount of each component needed in the blend product. The third set of constraints calculates the impurity parameter for each chemical, as a function of the various individual impurities. The fourth equation calculates the 'impurity parameter' for each blend product depending on the blending policy. The last set of constraints makes sure that all the impurity related specifications are satisfied by each blend product.

In sampling based algorithms, the expected cost is calculated using various realizations of the uncertain parameters (i.e. samples) and the corresponding treatment costs. The parameter $I_{il}(u)$ is then a function of each sample. The two stage stochastic programming blending problem is given as

First stage problem:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{k=1}^{p} C_i W_{ik} + E[R(W,\theta,u)] \tag{20}$$

where

$$\sum_{i=1}^{n} W_{ik} = \bar{W}_k, \quad \forall k = 1,\ldots,p \tag{21}$$

$$\sum_{i=1}^{n} x_{ij} W_{ik} \geq \bar{x}_{jk}, \quad \forall k = 1,\ldots,p \text{ and } j = 1,\ldots,m \tag{22}$$

Here, $E[R(W,\theta,u)]$ is the expected value of the recourse function, which is calculated in the second stage.

Second stage problem:

$$\text{Minimize} \quad E[R(W,\theta,u)] = \sum_{r=1}^{N_{\text{samp}}}\sum_{k=1}^{p} C_T \theta_k \tag{23}$$

where

$$\sum_{l=1}^{q} (I_{il}(r))^{\alpha_l} = I_i^*, \quad \forall i = 1,\ldots,n \tag{24}$$

$$\sum_{i=1}^{n} I_i^* W_{ik} = \bar{I}_k, \quad \forall k = 1,\ldots,p \tag{25}$$

$$\bar{I}_k.(1-\theta_k) \leq I_k^{\text{spec}}, \quad \forall k = 1,\ldots,p \tag{26}$$

The first stage decision variables are $W_{ik}$. The second stage considers various realizations of the uncertain parameters $I_{il}$ through $N_{\text{samp}}$ samples. This second stage problem minimizes the expected value of the recourse function through the decision variables $\theta_k$. This is a stochastic programming problem with a nonlinear relationship between the second stage parameters $I_{il}$ and $I_i^*$.

### 7.2. Simulations and results

This work considers the problem with seven chemicals ($A_1,\ldots,A_7$), three components ($C_1,\ldots,C_3$), 3 blend products ($P_1,\ldots,P_3$) and three different impurities, such as sulfur, ash and heavy residue, i.e. $n = 7$, $m = 3$, $p = 3$ and $q = 3$. The data for the problem is reported in Tables 2 and 3. $\alpha_1$, $\alpha_2$ and $\alpha_3$ are 0.9, 1.3, and 1.4, respectively, and the purification cost $C_T$ is US$ 10000 per unit reduction in the impurity. Each chemical has one uncertain impurity fraction. Here $I_{12}$, $I_{15}$, $I_{23}$, $I_{26}$, $I_{27}$, $I_{31}$ and $I_{34}$ are uncertain, varying by ±25% around the values reported in Table 2. All these uncertain parameters are normally distributed in the given range, i.e. ±25% range corresponds with the $\pm 3\sigma$ range, where $\sigma$ is the standard deviation of the normal distribution.

The problem is solved using the standard L-shaped algorithm and the proposed L-shaped BONUS algorithm, using both HSS and MCS techniques. In the L-shaped BONUS algorithm, the nonlinear relationship between $I_{il}$ and $I_i^*$ is bypassed using the reweighting scheme.

The optimal objective function value is plotted in Fig. 9 for different sample sizes. The results show that with the HSS technique, the average difference in the absolute values of the final objective function for the standard L-shaped and the L-shaped BONUS algorithm is only 1.6%, and this difference reduces

Table 3
Data for the blend products

| | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|
| $C_1$ fraction | 0.1 | 0.6 | 0.2 |
| $C_2$ fraction | 0.5 | 0.1 | 0.1 |
| $C_3$ fraction | 0.2 | 0.2 | 0.5 |
| Production (weight units) | 100 | 120 | 130 |
| $I_k^{\text{spec}}$ | 0.9 | 1.05 | 1.2 |

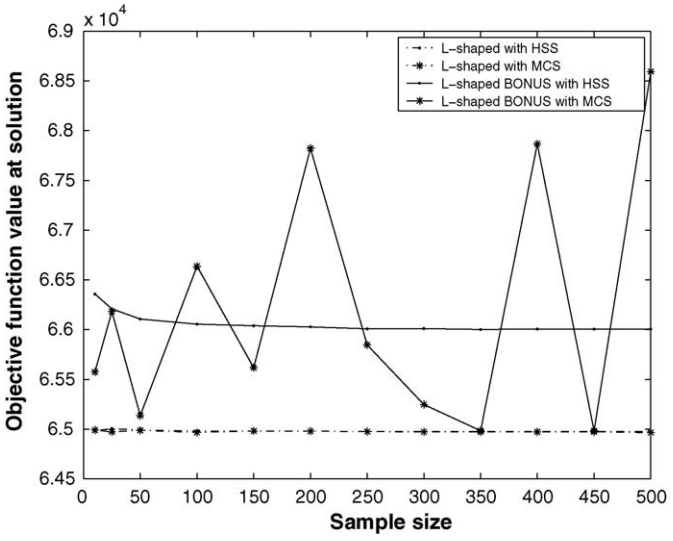


Fig. 9. Variation of the objective function value with sample size for the blending problem.

with the increasing sample size. Fig. 10 plots the number of iterations required to achieve the solution. It can be observed that the L-shaped algorithm consistently requires a larger number iterations. It is also observed that the L-shaped BONUS algorithm achieves an average reduction of 75% in the solution time over the standard L-shaped algorithm. This significant reduction, accompanied by a relatively small change in the final results, makes the L-shaped BONUS algorithm very attractive. A comparison between the results for the HSS and the MCS techniques throws up observations and conclusions similar to those for the farmer's problem. Thus, the MCS technique, in general, requires more iterations than the HSS technique, and results settle much faster with the HSS technique than with the MCS technique.

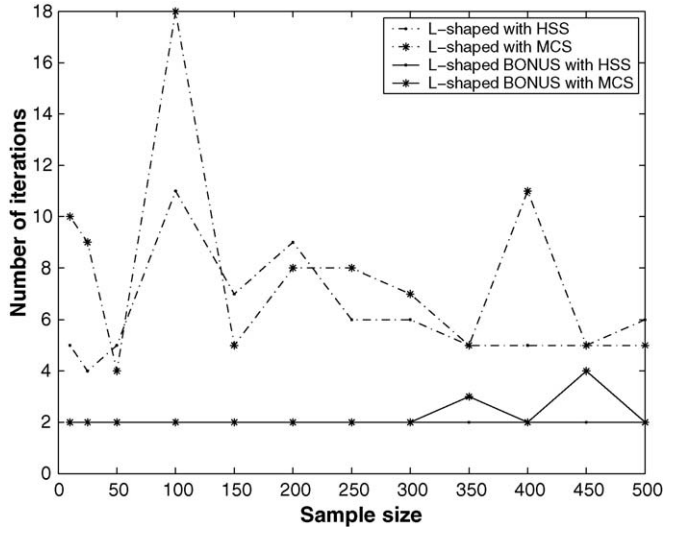


Fig. 10. Variation of number of iterations with sample size for the blending problem.

This section discussed the application of the proposed L-shaped BONUS algorithm to a process engineering related problem. Given the prevalence of the stochastic nonlinear programming problems in this field, the reported advantages of the L-shaped BONUS algorithm make it an important addition to the available solution techniques. The next section discusses a large scale real life problem of sensor placement in a water distribution network to further emphasize the advantages of this algorithm.

## 8. Sensor placement problem

Water pollution has a serious impact on all living creatures, and can negatively affect the use of water for drinking, household needs, recreation, fishing, transportation and commerce. The tragic events of September 11 have redefined the concept of risk and uncertainty, and thus water security has became a matter of utmost importance to national and international sustainability. In order to prepare for the catastrophic events, water utilities are feeling a growing need to detect and minimize the risk of malicious water contamination in distributed water networks. The problem assumes an interdisciplinary nature owing to the contributions from diverse fields such as, urban planning, network design and chemical contamination propagation. Hence a systems based effort initiated by the systems engineers, including the process systems engineers, is called for.

The integration of an adequate number of sensors at the appropriate places in a water distribution network can provide an early detection system, where appropriate control measures can be taken to minimize the risk. Since economics govern the maximum number of sensors available for this task, optimal utilization by placing them at the most appropriate locations in the network is essential. This results in an optimization problem. However, in order to obtain a robust solution in the face of risk, it is necessary to consider the various sources of uncertainty. This converts the deterministic optimization problem into a stochastic optimization problem. The next section explains the exact problem formulation.

### *8.1. Problem formulation*

The problem aims to find the optimal locations of a given number of sensors to minimize the cost and risk in the face of uncertain demands at various junctions. It is an SNLP problem since the relationship between the uncertain demands and the water network flow patterns is nonlinear.

The problem models a water network as a graph $G = (V, E)$, where $E$ is a set of edges representing pipes, and $V$ is a set of vertices, or nodes, where pipes meet (e.g. reservoirs, tanks, consumption points, etc.). An attack is modelled as the release of a large volume of a harmful contaminant at a single point in the network with a single injection. The water network simulator EPANET (Rossman, 1993) is used to determine an acyclic water flow pattern $p$, given a set of available water sources, assuming that each demand pattern $p$ holds steady for a sufficiently long time. The two stage stochastic mixed integer programming problem is given as

First stage problem:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}^{T} s_{ij} + E[R(c, s, u)]$$

where

$$s_{ij} = s_{ji}, \qquad i = 1, \ldots, n-1, i < j$$
$$\sum_{(i,j)\in E, i<j} s_{ij} \leq S_{\max}, \quad s_{ij} \in (0, 1); (i, j) \in E$$

Second stage problem:

$$\text{Minimize} \quad E[R(c, s, u)]$$
$$= \sum_{l=1}^{N_{\text{samp}}} \sum_{i=1}^{n} \sum_{p=1}^{P} \sum_{j=1}^{n} S\alpha_{ip}(l)\delta_{jp}(l)c_{ipj}$$

where

$$c_{ipi} = 1, \qquad i = 1, \ldots, n; p = 1, \ldots, P$$
$$c_{ipj} \geq c_{ipk} - s_{kj}, \quad (i, k, j) \in E; s.t. f_{kjp} = 1$$

Here, $\beta_{ij}$ is the cost of the sensor installed on branch $(v_i, v_j)$, $\alpha_{ip}$ is the probability of attack at node $v_i$, during flow pattern $p$, conditional on exactly one attack on a node during some flow pattern, $\delta_{jp}$ is the population density at node $v_j$ while flow pattern $p$ is active, and $c_{ipj}$ is the contamination indicator. $c_{ipj} = 1$ if node $v_j$ is contaminated by an attack at node $v_i$ during pattern $p$, and 0 otherwise. $s_{ij}$ is a binary variable indicating the sensor placement. It is 1 if a sensor is placed on an (undirected) edge $(v_i, v_j)$, and 0 otherwise. $S_{\max}$ is the maximum number of sensor allowed in the network. The risk is defined under a fixed number of flow patterns represented by the binary parameters $f_{ijp}$. $f_{ijp} = 1$ if there is a positive flow along (directed) edge $e = (v_i, v_j)$ during flow pattern $p$, and 0 otherwise. $S$ is the cost of a person getting affected by the contaminated water (such as treatment cost) and converts the risk into financial terms.

In the objective function, the first term gives the total cost of implementing sensors in the network, while the second term gives the expected cost due to the risk of contamination propagation. The uncertain flow demands with known probability distributions at the various nodes result in multiple flow patters $p$ in the network. It is necessary to consider all these patterns to simulate every possibility of the contamination propagation. This makes the problem stochastic. The uncertain space is characterized here through $N_{\text{samp}}$ samples in order to use the sampling based L-shaped BONUS algorithm.

The first stage problem uses $s_{ij}$ as the decision variables to minimize the total sensor cost and the expected risk that constitutes the recourse part of the problem. The constraint in the first stage problem restricts the maximum number of sensors used. The second stage problem minimizes the expected risk for $N_{\text{samp}}$ realizations of the uncertain demands. $c_{ipj}$ constitute the second stage decision variables. For this second stage problem, the first set of constraints ensures that when a node is directly attacked, it is contaminated. The second constraint set propagates the contamination from node $v_k$ to node $v_j$ if node $v_k$ is contaminated, there is positive flow along a directed edge from $v_k$ to $v_j$, and there is no sensor on that edge. See Shastri and Diwekar (2006), Berry, Fleischer, Hart, Phillips, and Watson (2005), Kessler, Ostfeld, and Sinai (1998) for the details of the problem formulation.

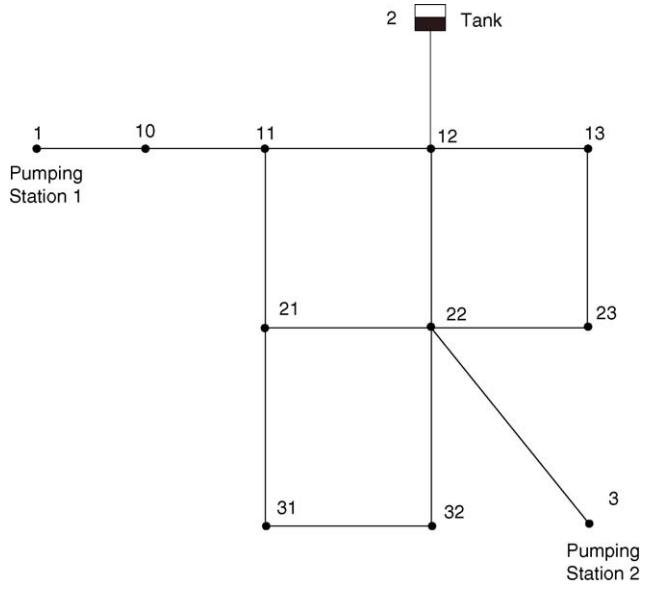


Fig. 11. Water distribution network for the sensor placement application of the L-shaped BONUS algorithm.

The network selected for the case study is a modification of the "Example Network 1" from EPANET (Rossman, 1999, 1993). The network is shown in Fig. 11. There are 12 nodes in the network, comprising of two pumping stations, one storage tank and nine consumptions points. Four nodes have uncertain demands, while the attack probability is considered to be fixed and equal at all the nodes. The network simulations generate the flow patterns giving the values of $f_{ipj}$, which are then used for the second stage problem solution.

This is a large scale optimization problem with 14 first stage decision variables, 1440 recourse variables and 1575 first and second stage constraints. The simulation requirements for such (and bigger) networks can be a quite high.

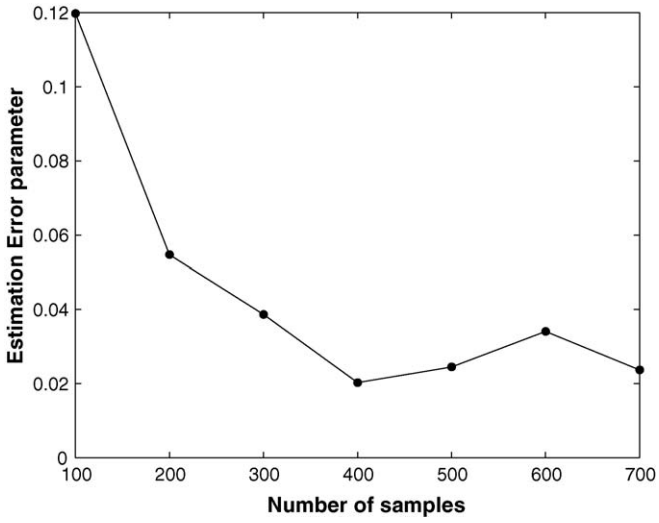


Fig. 12. Estimation error dependence on the number of samples.

Table 4
Comparative results of the different solution methods

| Maximum number of allowed sensors | Type of sensor | Deterministic analysis | | | | Stochastic analysis | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Estimated cost (US$) ($\times 10^7$) | Estimated percentage risk | Actual cost (US$) ($\times 10^7$) | Actual percentage risk | Estimated cost (US$) ($\times 10^7$) | Estimated percentage risk | Actual cost (US$) ($\times 10^7$) | Actual percentage risk |
| 1 | Low Cost | 2.1875 | 29.375 | 2.2860 | 32.364 | 2.2860 | 32.364 | 2.2860 | 32.364 |
| | High Cost | 2.4875 | 29.375 | 2.5860 | 32.364 | 2.5860 | 32.364 | 2.5860 | 32.364 |
| 2 | Low Cost | 1.8625 | 22.727 | 1.9914 | 25.628 | 1.9914 | 25.628 | 1.9914 | 25.628 |
| | High Cost | 2.4625 | 22.727 | 2.5914 | 25.628 | 2.5860 | 32.364 | 2.5860 | 32.364 |

### 8.2. Verification of the reweighting scheme

Before the problem solution and the results are reported, the validity of using the reweighting scheme for this model (EPANET water network) is ascertained by performing test simulations. The network was simulated for two different sets of samples, and the results for one set were estimated from the two sample sets using the reweighting scheme. These estimated results were then compared with the actual simulation results. It is observed that the estimation accuracy decreases with increasing number of uncertain variables. The effect of the samples size is shown in Fig. 12. The accuracy of estimation increases with the increase in the sample size, as the relative error between the estimated and the actual results goes down, as shown in Fig. 12.

### 8.3. Problem solution

The two stage problem posed above is solved using the internal sampling strategy, where the uncertain variables are sampled at the second stage problem solution, providing the statistical bounds on the recourse function. For comparative studies, a deterministic problem, formulated by ignoring uncertainties in the flow demands, is also solved using the deterministic solution technique.

In the standard sampling based L-shaped method, the EPANET will need to be simulated for each sample to get the flow pattern (giving the value of $f_{ijp}$), which is then used to solve the second stage dual problem. This is computationally very inconvenient, as it requires to link the optimization code with the EPANET simulation software and run the simulation for each sample.

The proposed algorithm simplifies the task by following the path shown in Fig. 5. So the EPANET is first simulated for a certain number of samples to get the distribution of the various flow patterns. The number of samples is decided by the desired accuracy. For this work, 100 samples were used. Then, for every next iteration, the reweighting scheme is used to estimate the distribution of the flow patterns. This algorithm does not need to connect the optimization code with the EPANET simulator, since the initial simulations can be done off-line, and the results can be stored and used by the optimization code.

### 8.4. Results

Some of the important and representative results for the sensor placement problem are given in Table 4. The table reports the estimated and the actual objective function values and the percentage risk values for both the analysis. The estimated values are obtained from the particular problem solution, while the actual values are obtained through stochastic simulations. Since the stochastic analysis incorporates the stochastic simulation results in decision making, the estimated and the actual values for this method are same.

A comparison of the results shows that the estimated values for the two methods are different, those for the deterministic method being lower than those for the stochastic method. The actual objective function values and the percentage risk values are however more important for comparing the different solutions. These actual values are higher for the deterministic solution than for the stochastic solution. These results point to the fact that the results from deterministic analysis are clearly suboptimal and the consideration of uncertainty is important in this problem. Without the consideration of uncertainty, the problem will be deterministic, not needing the proposed algorithm for its solution. But the importance of uncertainty is manifested by the actual risk and the objective function values in Table 4, as well as the placement locations of the sensors (not shown in the representative table here). The stochastic problem would have been computationally highly demanding had it been solved by the traditional methods. But the proposed algorithm improves the performance and makes the solution possible with considerable ease. This is the important result from this case study.

## 9. Conclusion

The paper proposes a new algorithm to solve the stochastic nonlinear programming problems. The SNLP problems, being computationally demanding in most cases, have found little application. The proposed L-shaped BONUS algorithm overcomes the computational hurdle by using reweighting in the traditional sampling based L-shaped algorithm structure. The reweighting scheme has been successfully employed in a recently proposed BONUS algorithm, which also solves the SNLP problems. The combination of the decomposition structure from the L-shaped method, the reweighting scheme from the BONUS algorithm and the efficient Hammersley Sequence Sampling technique, makes the proposed L-shaped BONUS algorithm computationally highly efficient. The results for the case study problems, a well known farmer's problem, a process systems engineering relevant chemical blending problem, and a sensor placement problem in a water distribution security network show that the algorithm is a valuable tool in solving the SNLP problems with

a considerably reduced computational burden. In all the cases, the reweighting approximation is shown not to severely compromise the accuracy while greatly reducing the computation times. It was also shown for the first two problems that the HSS technique is better than the MCS technique for this sampling based algorithm. The sensor placement problem is particularly interesting as it is a large scale application in an emerging area of water security. It is a computationally expensive problem for the traditional two stage algorithm. The proposed L-shaped BONUS algorithm thus holds considerable promise and needs to be investigated further to identify the additional properties and application areas.

## Acknowledgements

This work is funded by the National Science Foundation under the grant CTS-0406154.

## References

Berry, J., Fleischer, L., Hart, W., Phillips, C. & Watson, J. (2005). Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, *131*, 237–243.

Birge, J., & Louveaux, F. (1997). *Introduction to stochastic programming*, Springer series in Operations Research.

Charnes, A., & Cooper, W. (1959). Chance-constrained programming. *Management Science*, *5*, 73–79.

Dantzig, G., & Glynn, P. (1990). Parallel processors for planning under uncertainty. *Annals of Operations Research*, *22*, 1–21.

Dantzig, G., & Infanger, G. (1992). *Computational and applied mathematics I*. Elsevier Science Publishers, B.V. (North Holland).

Diwekar, U. (2003a). Greener by design. *Environmental Science and Technology*, *37*, 5432–5444.

Diwekar, U. (2003b). *Introduction to applied optimization*. Dordrecht: Kluwer Academic Publishers.

Diwekar, U. (2003c). Optimization under uncertainty in chemical engineering. *Proceedings of Indian National Science Academy*, *69A* (3–4), 267–284.

Diwekar, U. (2005). Green process design, industrial ecology, and sustainability: A systems analysis perspective. *Resources, Conservation and Recycling*, *44* (3), 215–235.

Elmaghraby, S. (1960). Allocation under uncertainty when the demand has continuous D.F. *Management Science*, *10*, 270–294.

Gray, F. (1986). *Petroleum production for the nontechnical person*. Tulsa, OK: PennWell Books.

Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distribution. *Technometrics*, *37*, 185–194.

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

Jung, J., Blau, G., Penky, J., Reklaitis, G., & Eversdyk, D. (2004). A simulation based optimization approach to supply chain management under demand uncertainty. *Computers and Chemical Engineering*, *28*, 2087–2106.

Kalagnanam, J., & Diwekar, U. (1997). An efficient sampling technique for off-line quality control. *Technometrics*, *39* (3), 308–319.

Kessler, A., Ostfeld, A., & Sinai, G. (1998). Detecting accidental contaminations in municipal water networks. *Journal of Water Resources Planning and Management*, *124*, 192–198.

Kheawhom, S., & Hirao, M. (2004). Decision support tools for environmentally benign process design under uncertainty. *Computers and Chemical Engineering*, *28*, 1715–1723.

Lin, X., Janak, S., & Floudas, C. (2004). A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Computers and Chemical Engineering*, *28*, 1069–1085.

Louveaux, F. (1978). Piecewise convex programs. *Mathematical Programming*, *15*, 53–62.

Louveaux, F. (2001). Nonlinear stochastic programming. Technical report, Faculté des Sciences, Département de Mathématique.

Rhodes, A. (1993). Refinery operating variables key to enhanced lube oil quality. *Oil and Gas Journal*, *January*, 45.

Rockafellar, R., & Wets, R. -B. (1986). A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming. *Mathematical Programming Study*, *28*, 63–93.

Rockafellar, R., & Wets, R. -B. (1991). Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, *16*, 119–147.

Rossman, L. (1993). *EPANET Users Manual*. Cincinnati, OH: Risk Reduction Engineering Laboratory, Environmental Protection Agency.

Rossman, L. (1999). The EPANET programmer's toolkit for analysis of water distribution systems. In *Proceedings of the Annual Water Resource Planning and Management Conference*.

Ruszczyński, A. (1986). A regularized decomposition for minimizing a sum of polyhedral functions. *Mathematical Programming*, *35*, 309–333.

Rya, J. -H., Dua, V., & Pistikopoulos, E. (2004). A bilevel programming framework for enterprise-wide process networks under uncertainty. *Computers and Chemical Engineering*, *28*, 1121–1129.

Sahin, K., & Diwekar, U. (2004). Better optimization of nonlinear uncertain systems (BONUS): A new algorithm for stochastic programming using reweighting through kernel denisty estimation. *Annals of Operations Research*, *132*, 47–68.

Sahinidis, N. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering*, *28*, 971–983.

Shastri, Y., & Diwekar, U. (2006). Sensor placement in water networks: A stochastic programming approach. *Journal of Water Resources Planning and Management*, in press.

Silvermann, B. (1986). *Density estimation for statistics and data analysis*. Boca Raton, USA: Chapman and Hall CRC reprint 1998.

Van Slyke, R., & Wets, R. -B. (1969). L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, *17*, 638–663.

Wang, R., Diwekar, U., & Grégorie Padró, E. (2004). Efficient sampling techniques for uncertainties and risk analysis. *Environmental Progress*, *23* (2), 141–157.

Ziemba, W. (1970). Computational algorithms for convex stochastic programs with simple recourse. *Operations Research*, *18*, 414–431.

Ziemba, W. (1972). Solving nonlinear programming problems with stochastic objective functions. *The Journal of Financial and Quantitative Analysis*, *7* (3), 1809–1827.