Efficient Optimization Algorithm for Large Scale Problems in Nonlinear Stochastic Programming

Y. Shastri
Department of Bioengineering, University of Illinois at Chicago and
Vishwamitra Research Institute, 34, N. Cass Avn., Westmont, IL 60559
U. Diwekar
Vishwamitra Research Institute,
34, N. Cass Avn., Westmont, IL 60559

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

1 Introduction

Stochastic nonlinear programming (SNLP) problems represent an important class of 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 range or, in some cases, in terms of their probability distributions. In such cases, stochastic programming methods need to be resorted to for optimization.

The field of process systems engineering is also replete with 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 stochastic programming methods and their applications in process engineering field is given in [1] and [2]. Some of the recent applications include enterprise-wise process network [3], planning and scheduling related tasks [4, 5] and environment related applications [6, 7, 8]. Many of these problems are nonlinear complicating the problem solution.

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

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

where, θ is the decision variable, x is the set of system parameters, u is the set of uncertain variables and P_1, P_2 and P_3 are probabilistic measures such as 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 available solution algorithms, high computational requirements being one of them. This work proposes a new algorithm, the L-shaped BONUS, to solve 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. This new 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 SLP (stochastic linear programming) 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 new BONUS algorithm along with the reweighting scheme, 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, 7 and 8 give details of algorithm steps and computational advantages through various case study problems. The final section draws concluding remarks.

2 Methods for SNLP problem solution

2.1 General overview

Over the years, a lot of research has gone into devising strategies to solve the SNLP problems. One kind of solution methods, such as the chance constraint programming method [9], convert these problems into deterministic equivalents. Deterministic optimization methods can then be applied. But these methods are restricted to 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 [10, 11, 12]. For optimization problems than can be decomposed into two or multiple stages, decomposition based stochastic programming methods such as the L-shaped method are developed [13]. The basic L-shaped method is modified

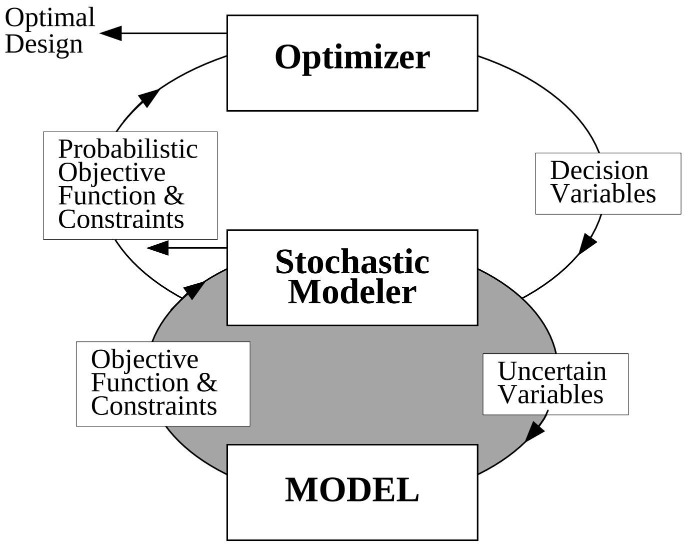

Figure 1: Optimization under Uncertainty

for different problem types, which includes regularized decomposition method [14] and piecewise quadratic form of L-shaped method [15]. Methods based on the stochastic programming Lagrangian include the basic Lagrangian dual ascent method [13], the Lagrangian finite generation method for linear quadratic stochastic programs [16] and the progressive hedging algorithm [17]. These decomposition based methods require convexity of the problem and/or dual block angular structure. The stochastic quasi-gradient methods (SQG) are less specialized than the other algorithms but are useful to solve problems with complex objective functions and constraints [18]. 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 respective references made for these solution algorithms. Even though important, application of these methods to solve real life problems has always been restricted. This is because of 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 uncertain parameters are known. The aim is to model the complete uncertain parameter space as closely as possible through sufficient number of samples. Figure 1 represents the generalized solution procedure for 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 with each iteration. In the set up of figure 1, calculation of these terms needs simulation of the stochastic modeler at each iteration. In traditional sampling based methods, this is achieved by model simulations for given number of samples and subsequent computation of the probabilistic function (e.g. expected value of the objective function). Two such methods are the sampling based L-shaped method [19, 20] and Stochastic Decomposition algorithm [21]. L-shaped method is a scenario based method. But the number of scenarios increase exponentially as the number of uncertain variables increase. 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.

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

3 BONUS

Sampling based approaches to solve 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 SNLP problems which holds its advantage by circumventing the problem of repeated model simulations. This new method is Better Optimization of Nonlinear Uncertain Systems (BONUS) [22]. Figure 2 shows the BONUS algorithm structure.

Figure 1 has shown the standard stochastic programming algorithm structure with the sampling loop requiring repeated model simulations. Compared to this, BONUS algorithm uses 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 objective function, new set of samples are taken. But this time the model is not re-run, instead reweighting approach is applied to approximate the probabilistic behavior of the new output distribution. Figure 2 illustrates this step of the algorithm. The reweighting scheme uses the initial sample set, initial output distribution and new sample set data to estimate information about the new output distribution. Owing to the importance of reweighting in the proposed algorithm, it will be explained in the following section.

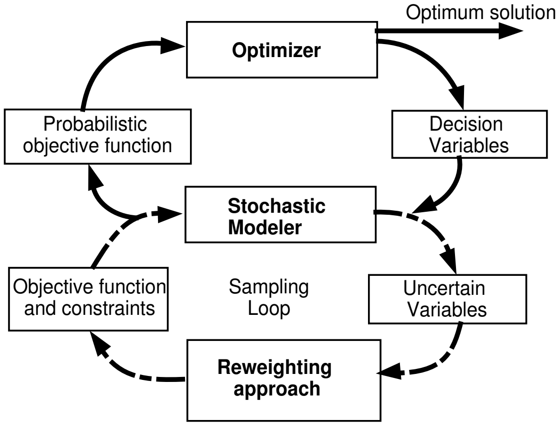

Figure 2: Bonus Algorithm

3.1 The reweighting approach

Reweighting approach is based on the various reweighting schemes proposed in [23]. 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 respective probability density functions. Let X be a random variable with 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 problem of estimating $E_{\hat{f}}[Y(X)]$, where

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

and samples X_i are now drawn from $\hat{f}(x)$. Distribution $\hat{f}(x)$ can be designed to achieve desired results (e.g. reduced variance, better representation of 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 target and design distributions and weighs observations of $Q(x)$. To perform this estimation effectively, Hesterberg [23] proposed various design distributions $\hat{f}(x)$ (e.g. defensive mixture distributions) and estimation schemes (integration estimate, ratio estimate). In ratio estimate, weights W_i are normalized to avoid problems when they do not sum to 1.

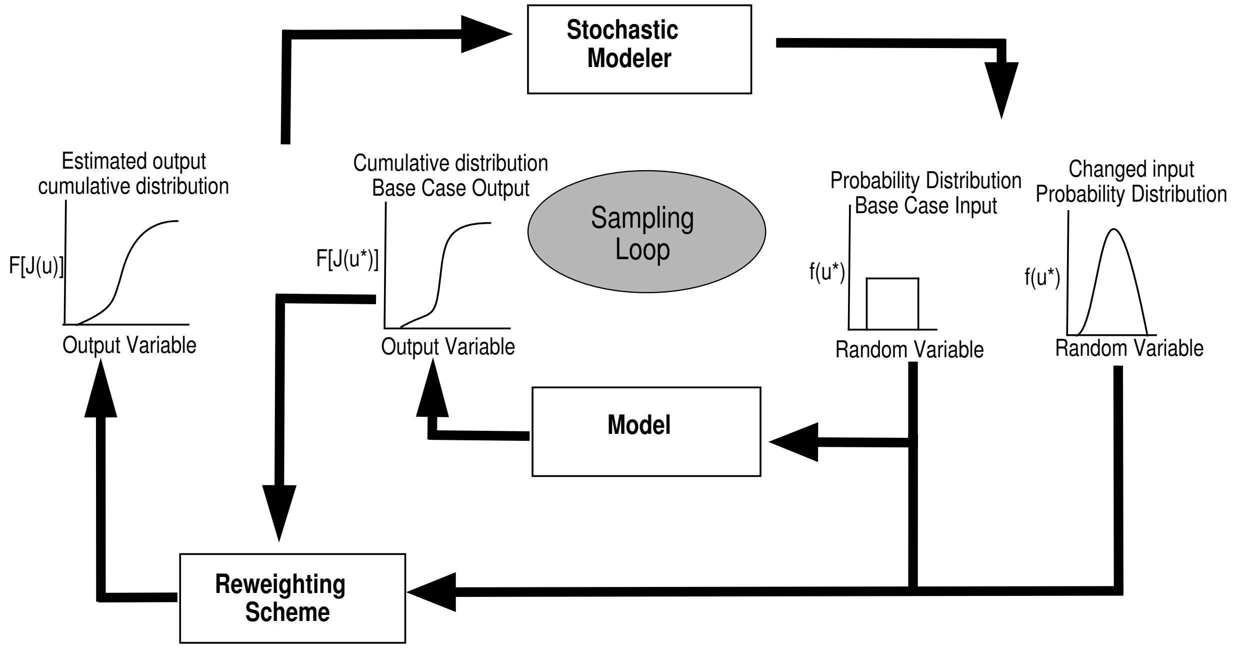
Stochastic Modeler
Estimated output cumulative distribution
F[J(u)]
Output Variable
Cumulative distribution Base Case Output
F[J(u*)]
Output Variable
Sampling Loop
Probability Distribution Base Case Input
f(u*)
Random Variable
Changed input Probability Distribution
f(u*)
Random Variable
Model
Reweighting Scheme

Figure 3: Reweighting approach in BONUS

The normalized weights V_i and estimate μ 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 figure 3. Suppose X represents the uncertain variable in stochastic programming problem and $Q(X)$ is the output of stochastic modeler. For the first iteration, base case samples X_i^* with 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 required distribution ($f(x)$) are drawn. Having known the model response $Q(X_i^*)$ for sample set X_i^* from distribution $\hat{f}(x)$, it is possible to use equation 4 to estimate the expected value of model response $Q(X_i)$ for new sample set X_i from distribution $f(x)$. The expected value of the stochastic model response $Q(X_i)$ for new sample set X_i is therefore given as

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

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 [24] which is a nonparametric density estimation technique. The basic idea behind this technique is to place a bin of 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 normal density function, the density function for the sample set X_i is calculated using equation 6.

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

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

Thus, given two sample sets, equation 6 is used to determine the density function at each sample point for both the distributions which are then used in equation 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 decomposable SNLP problems in computationally efficient manner. But before describing the proposed algorithm, it is prudent to understand the L-shaped method which is explained in next section.

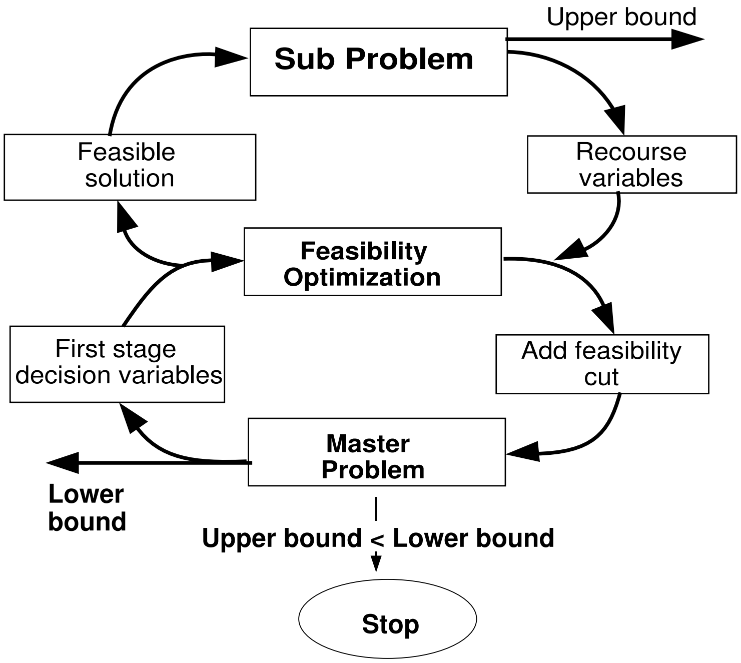

Figure 4: L-shaped algorithm structure

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 [25]. 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 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. Figure 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 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 [26].

For continuous distributions, sampling is used to approximate the distribution. Use of importance sampling as a variance reduction technique was proposed in [19] while use of sampling in the basic L-shaped method was proposed in [20]. 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 context of the sampling based algorithms explained in section 2, sub-problem solution for each realization of uncertain space can be compared with the stochastic modeler in figure 1.

Thus to summarize, decomposition strategy of L-shaped method offers computational savings but is still not very efficient for the SNLP problems. Therefore, we are proposing integration of decomposition structure and 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 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, mathematical details are not reproduced here and can be found in sections 3 and 4. The proposed algorithm is shown schematically in figure 5 and explained below.

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 made using a linearized approximation of the second stage nonlinear recourse function and utilizing the feasibility and optimality cuts, if generated. This also determines the lower bound for the objective function. The second stage objective is the expected value of the recourse function, which depends on the first as well as 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 computations at the sub-problem solution stage by using reweighting scheme to bypass nonlinear model computations. The reweighting scheme, as mentioned in section 3, needs the model output distribution for base case uniform input distribution. For this purpose, during the first optimization iteration, the nonlinear model is simulated and sub-problem solved for each sample. Model simulation results for the base case constitute the base case output distribution. 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 as per the L-shaped algorithm. 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 new set of samples are taken by the stochastic modeler, model simulation and sub-problem solution is 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 cuts and update the objective function upper bound. It should thus be noted that for second iteration, only one sub-problem is solved. So not only the nonlinear model simulation time but also the sub-problem solution

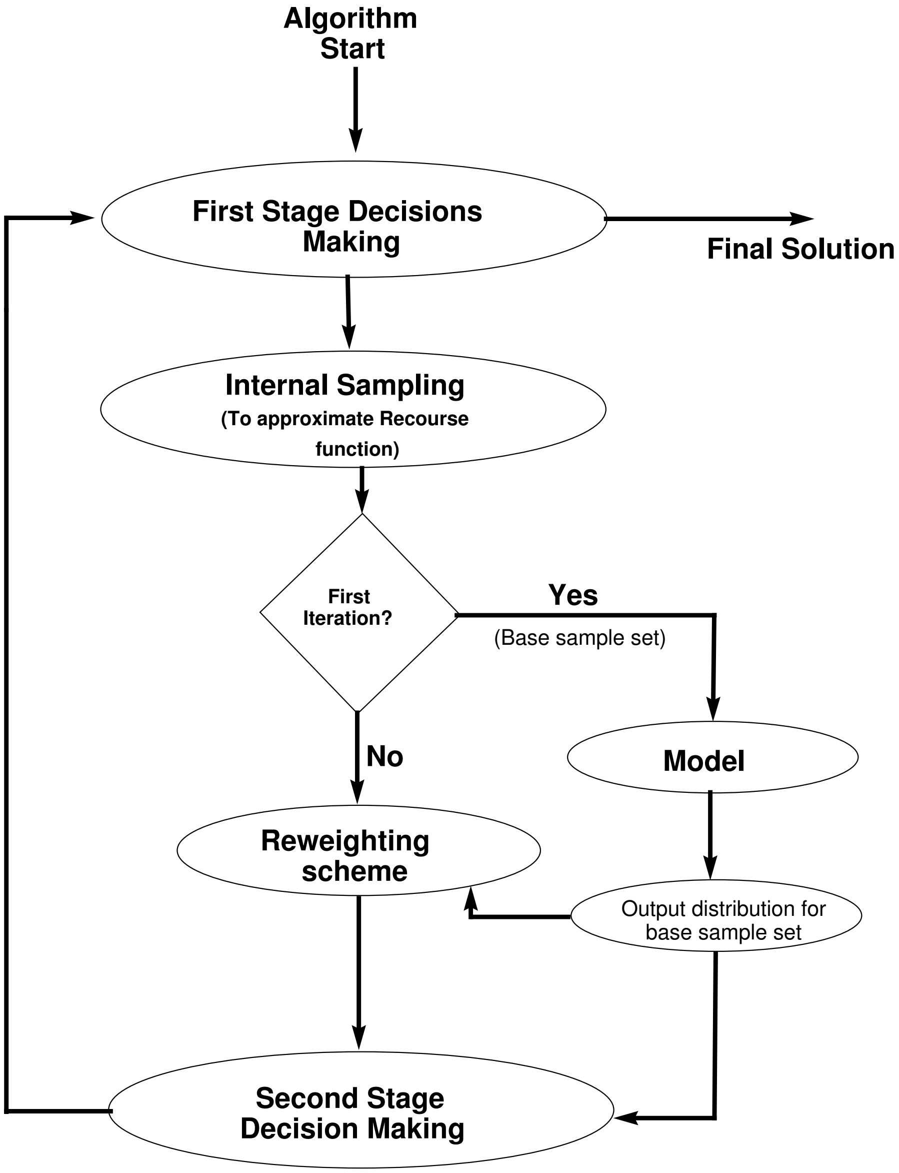
Algorithm Start
First Stage Decisions Making
Final Solution
Internal Sampling
(To approximate Recourse function)
First Iteration?
Yes
(Base sample set)
No
Model
Reweighting scheme
Output distribution for base sample set
Second Stage Decision Making

Figure 5: The proposed L-shaped BONUS algorithm structure

time is saved. This procedure of reweighting based estimation is continued in every subsequent iteration till the L-shaped method based termination criteria 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 stochastic optimization procedure, being avoided, problem solution becomes faster. The effect is expected to be more pronounced in case of nonlinear and/or high dimensional models, as the ones often encountered for real life systems. Another advantage of this algorithm is its ability to convert an SNLP problem into an SLP problem by using reweighting to approximate 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 estimation accuracy of reweighting improves with 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 details readers are referred to [22]

Finally, as with any sampling based optimization technique, sampling properties are very important for this algorithm. The accuracy of the reweighting scheme depends on the number and uniformity of samples (see section 8). For this algorithm, we propose to use the Hammersley Sequence Sampling (HSS) which is shown to be very efficient [27, 28]. The sampling technique is based on the generation and inversion of 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 [13]. The problem, as formulated in [13], is a stochastic linear programming problem which is modified into an SNLP problem.

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 tones (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 selling cost due to wholesaler’s margin and transportation cost. Sugar beet sells at a cost of $36/T if the amount is less than 6000 T. Any additional quantity can be sold at $10/T only. Through experience the farmer knows that the mean yield of crops is 2.5 T, 3 T 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 uncertain yields. Table 1 summarizes the data and more details about the SLP can be found in [13].

For this illustration, to convert the problem into an SNLP problem, the uncertain yield is assumed to be dependent on four different factors which are uncertain. These four factors are the average rainfall, availability of sunlight, attack probability of a crop disease and the probability of

Table 1: Data for farmer's problem

	Wheat	Corn	Sugar Beets
Yield (T/acre)	2.5	3.0	20
Planting cost ($/acre)	150	230	260
Selling price ($/T)	170	150	36 under 6000 T 10 above 6000 T
Purchase price ($/T)	238	210	-
Minimum requirement (T)	200	210	-

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:

$$\begin{aligned} Y_r &= 2\,\alpha_r\,(1-\frac{\alpha_r}{2}) \qquad \alpha_r \in [0,2] && (7)\\ Y_s &= 1.58\,(1-e^{-\alpha_s}) \qquad \alpha_s \in [0,1] && (8)\\ Y_d &= 1-\alpha_d \qquad \alpha_d \in [0,1] && (9)\\ Y_p &= 1-\alpha_p^2 \qquad \alpha_p \in [0,1] && (10) \end{aligned}$$

where,

- Y_i are fractions of the maximum yield due to corresponding effects
- α_r: Fractional rainfall of the yearly average
- α_s: Fractional sunlight of the yearly average
- α_d: Attack probability of a crop disease
- α_p: Attack probability of pests

The overall fractional yield of the crops is given by

$$Y_{actual} = Y_r \times Y_s \times Y_d \times Y_p \times Y_{max} \qquad (11)$$

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:

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

subject to the following constraints

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

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

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

First Stage Problem

$$\begin{array}{rrcl} \text{Min} & 150x_1 & + & 230x_2 + 260x_3 + \theta \\ \text{s.t.} & x_1 + x_2 + x_3 & \leq & 500, \\ & G_l\, x + \theta & \geq & g_l \qquad l = 1 \dots s, \\ & x_1, x_2, x_3 & \geq & 0, \end{array}$$

where θ 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 optimality cuts applied during iterations of the L-shaped method.

Second Stage Problem

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

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 sampling based L-shaped method, involves dual formulation of the nonlinear second stage problem and 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 dual problem solutions for the samples can be demanding.

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

The exact solution procedure is as follows. At every second stage problem solution, 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 crop yield and the yield value is used to solve the dual for each sample (i.e. n dual problem solutions) and optimality cut, if needed, is generated. The first sample set is stored as the base sample set.

At subsequent iterations, during the second stage solution, the new set of n samples are taken and 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. 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 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

$$\begin{array}{rrcl} \text{Min} & 150x_1 & + & 230x_2 + 260x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 & \leq & 500 \\ & x_1, x_2, x_3 & \geq & 0 \end{array}$$

  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 crop using the n sampled uncertain variables and relations 7 to 10.
- Step 4: Solve the following dual problem for the n samples of crop yield. The values of x_i^1 are passed on to the second stage.

$$\begin{array}{rrcl} \text{Max} & \pi_1(200 - Y_1\, x_1^1) & + & \pi_2(240 - Y_2\, x_2^1) - \pi_3(Y_3\, x_3^1) - 6000\pi_4 \\ \text{s.t.} & \pi_1 & \leq & 238 \\ & \pi_2 & \leq & 210 \\ & \pi_1 & \geq & 170 \\ & \pi_2 & \geq & 150 \\ & \pi_3 + \pi_4 & \geq & 36 \\ & \pi_3 & \geq & 10 \\ & \pi_1, \pi_2, \pi_3, \pi_4 & \geq & 0 \end{array} \qquad (12)$$

The solution of problem 12 for 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 $w > \theta$, optimality cut is introduced.

Iteration 2:

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

$$\begin{array}{rrcl} & \text{Min} \quad 150x_1 & + & 230x_2 + 260x_3 + \theta \\ \text{s.t.} & x_1 + x_2 + x_3 & \leq & 500 \\ & \theta & \geq & 98000 - [610.1 \;\; 636.4 \;\; 727.4][x_1 \;\; x_2 \;\; x_3]^T \\ & x_1, x_2, x_3 & \geq & 0 \end{array} \tag{13}$$

  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 $\{\text{u}\}$
- Step 3: Calculate the estimated yield of the crops using the base and new sample sets bypassing relations 7 to 10. The estimated yield is 0.842.
- Solve the dual problem given by 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 solutions 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 $w \leq \theta$ is satisfied.

6.3 Results of the farmer's problem

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

Based on these plots HSS emerges as a more efficient sampling technique than MCS. The results for HSS appear to reach the steady state value faster than for MCS as the sample size is increased. This claim is further corroborated by figure 8 which plots the iterations needed to reach the solution for different sample sizes. It can be observed that for standard L-shaped method, MCS sampling technique needs more iterations in general than 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 variance of a derived distributions by the HSS method is on average 3 to 100 times less than the MCS and other stratified sampling techniques [27]. For the same sample size, the HSS method therefore approximates a given distribution better than the MCS. This results in faster convergence of HSS based algorithms in general. For the proposed algorithm though, both, MCS and HSS sampling techniques need 6 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 sample size and sampling method changes. But better values of final solutions (figures 6 and 7) confirm the superiority of the HSS method over MCS.

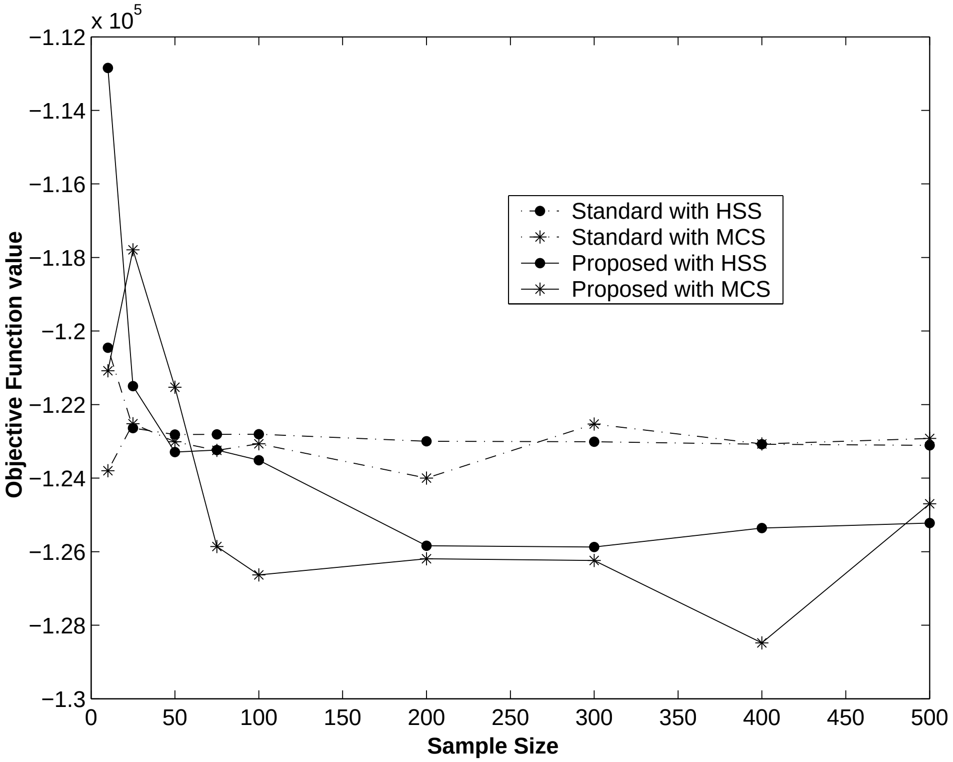

Figure 6: Variation of objective function with sample size for farmer's problem

Computational time is an important factor while comparing these algorithms. Computational time increases exponentially with sample size for the standard L-shaped method while it increases almost linearly for the proposed L-shaped BONUS algorithm. The computational efficiency of L-shaped BONUS therefore becomes more pronounced as the sample size is increased. With the need to increase sample size to improve 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. 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 crude oils by distillation. The additives are chemicals used to give the base stocks

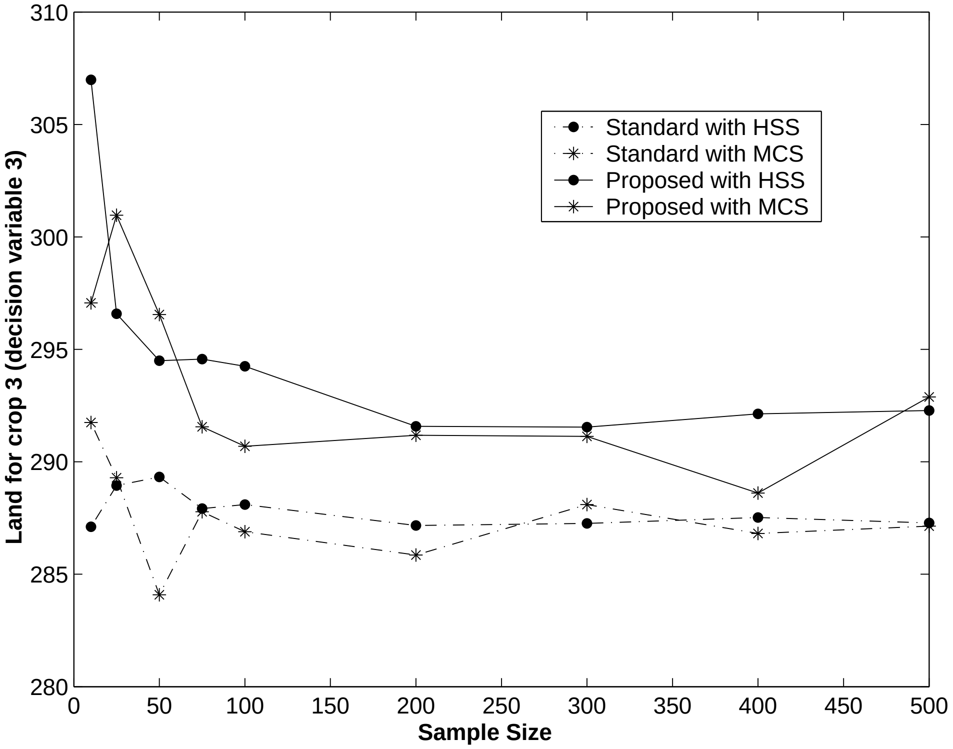

Figure 7: Variation of decision variable 3 with sample size for farmer’s problem

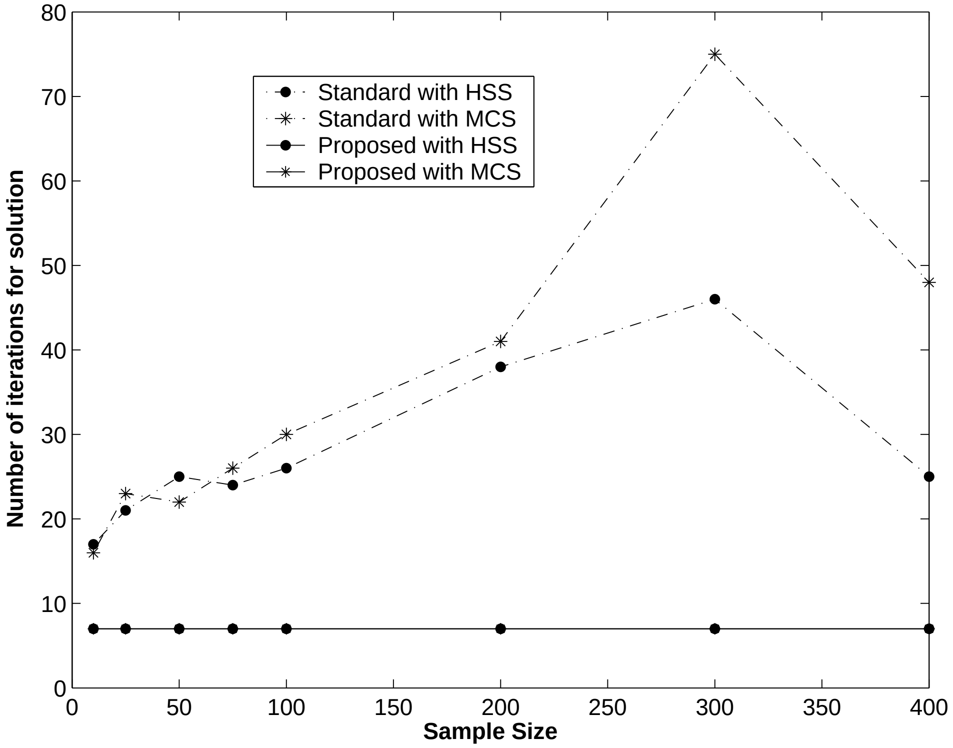

Figure 8: Variation of iteration requirement with sample size for farmer’s problem

desirable characteristics which they lack of to enhance and improve existing properties [29, 30]. In the context of such an application, a general chemical blending optimization problem is explained below followed by 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 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.). Market demands call for production of a particular quantity of each blend product. Blend products catering to different applications (e.g. high performance lube oil, grease, industrial grade lube oil etc.) have different specifications on fractions of m different components (for a lube oil such specifications will depend on physical property requirements like pour point, viscosity, boiling temperature). These specifications need to be satisfied to market the blend products. The task is complicated due to the presence of q impurities in the chemicals. Exact mass 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 product of other processes (such as crude distillation for lube oil base stocks). There are also 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 impurities below specifications. The treatment cost depends on the amount of reduction in the impurities to be achieved. The goal in formulating the stochastic optimization problem is to find the optimum blend policy to minimize raw material cost and expected blend product treatment cost in the presence of uncertainty associated with impurity content of the chemicals. The stochastic programming problem is formulated as below.

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

Subject to:

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

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

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

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

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

Here, W_{ik} is the 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 of blend product k. x_{ij} is the fraction of component j in chemical i and $\bar{x}_{jk}$ is the specification of component j in blend product k. $I_{il}(u)$ is the (possibly uncertain) 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 various impurities. Coefficients α_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^{spec} is the maximum permitted impurity content in the blend product. θ_k is the purification required for blend k to satisfy the impurity constraint.

The objective function consists of two parts. The first part is the cost of 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. Second constraint set ensures that 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. Third set of constraints calculates the impurity parameter for each chemical, as a function of 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 uncertain parameters (i.e. samples) and the corresponding treatment costs. Parameter $I_{il}(u)$ is then a function of each sample. The two stage stochastic programming blending problem is given as

First stage problem

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

where

$$\begin{aligned} \sum_{i=1}^{n} W_{ik} &= \bar{W}_k \qquad \forall k = 1,\ldots,p & (21)\\ \sum_{i=1}^{n} x_{ij} W_{ik} &\geq \bar{x}_{jk} \qquad \forall k = 1,\ldots,p \text{ and } j = 1,\ldots,m & (22)\end{aligned}$$

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

Second stage problem

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

where

$$\begin{aligned} \sum_{l=1}^{q} (I_{il}(r))^{\alpha_l} &= I_i^* \qquad \forall i = 1,\ldots,n & (24)\\ \sum_{i=1}^{n} I_i^* W_{ik} &= \bar{I}_k \qquad \forall k = 1,\ldots,p & (25)\\ \bar{I}_k.(1-\theta_k) &\leq I_k^{spec} \qquad \forall k = 1,\ldots,p & (26)\end{aligned}$$

The first stage decision variables are W_{ik}. The second stage considers various realizations of uncertain parameters I_{il} through N_{samp} samples. This second stage problem minimizes the expected value

Table 2: Data for chemicals in 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 ($/unit weight)	104	90	135	130	115	126	120

Table 3: Data for 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^{spec}	0.9	1.05	1.2

of the recourse function through decision variables θ_k. This is a stochastic programming problem with a nonlinear relationship between second stage parameters I_{il} and I_i^*.

7.2 Simulations and results

This work considers the problem with 7 chemicals ($A_1, \ldots, A_7$), 3 components ($C_1, \ldots, C_3$), 3 blend products ($P_1, \ldots, P_3$) and 3 different impurities, such as sulfur, ash and heavy residue, i.e. $n = 7$, $m = 3$, $p = 3$ and $q = 3$. Data for the problem is reported in tables 2 and 3. α_1, α_2 and α_3 are 0.9, 1.3, and 1.4, respectively and the purification cost C_T is $ 10000 per unit reduction in impurity. Each chemical has one uncertain impurity fraction. Here I_{12}, I_{15}, I_{23}, I_{26}, I_{27} and I_{34} are uncertain, varying by $\pm 25\%$ around the values reported in table 2. All these uncertain parameters are normally distributed in the given range, i.e. $\pm 25\%$ range corresponds with the $\pm 3\sigma$ range where σ is the standard deviation of normal distribution.

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

The optimum objective function value is plotted in figure 9 for different sample sizes. The results show that with HSS technique, average difference in the absolute values of the final objective function for the standard L-shaped and L-shaped BONUS algorithm is only 1.6%, and this difference reduces with increasing sample size. Figure 10 plots the number of iterations required to achieve the solution. It can be observed that the L-shaped algorithm consistently requires more iterations. It is also observed that the L-shaped BONUS algorithm achieves an average reduction of 75% in solution time over standard L-shaped algorithm. This significant reduction accompanied by a relatively small change in the final results makes L-shaped BONUS algorithm very attractive. Comparison between the HSS and MCS techniques throws up observations and conclusions similar to those for the farmer's problem. Thus MCS technique in general requires more iterations than HSS technique

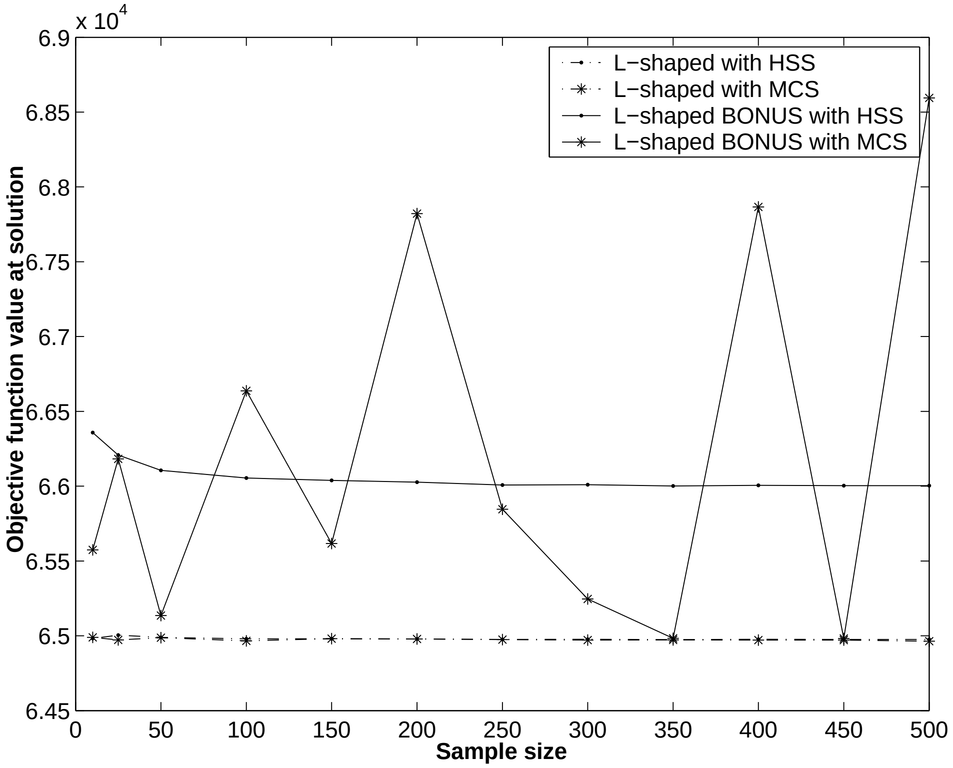

Figure 9: Variation of objective function with sample size for blending problem

and results with HSS settle much faster than with MCS.

This section discussed the application of the proposed L-shaped BONUS algorithm to a process engineering related problem. Given the prevalence of stochastic nonlinear programming problems in this field, the reported advantages of the L-shaped BONUS algorithm make this 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, thus water security has became a matter of utmost importance to national and international sustainability. In order to prepare for 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 interdisciplinary nature owing to contributions from diverse fields such as, urban planning, network design and chemical contamination propagation. Hence a systems based effort initiated by systems engineers, including process systems engineers, is called for.

Integration of adequate number of sensors at 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,

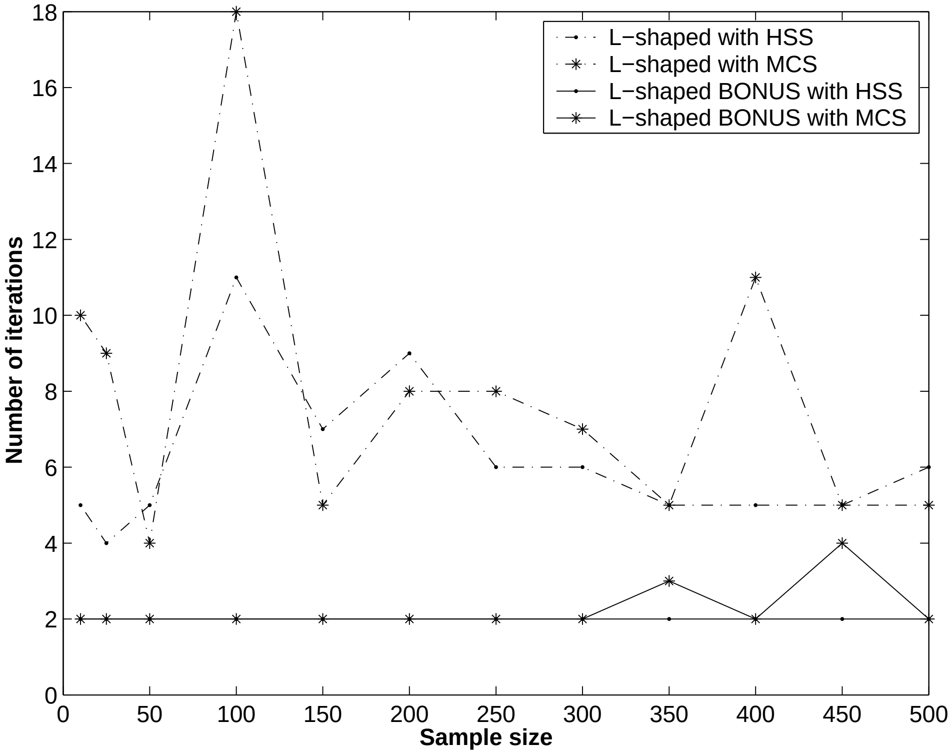

Figure 10: Variation of iteration requirement with sample size for blending problem

optimal utilization by placing them at the most appropriate locations in the network is essential resulting in an optimization problem. However, in order to obtain a robust solution in the face of risk, it is necessary to consider 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 optimum locations of a given number of sensors to minimize cost and risk in the face of uncertain demands at various junctions. It is an SNLP problem since the relationship between uncertain demands and 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 [31] is used to determine an acyclic water flow pattern p, given a set of available water sources, assuming 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

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

$$\begin{aligned} & where \\ & s_{ij} \;=\; s_{ji} && i=1,\ldots,n-1,\; i \;<\; j \\ & \sum_{(i,j)\in E, i<j} s_{ij} \;\leq\; S_{max} && s_{ij} \in (0,1); (i,j) \in E \end{aligned}$$

Second Stage Problem

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

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

Here β_{ij} is the cost of sensor installed on branch (v_i, v_j), α_{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, δ_{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 sensor placement. It is 1 if a sensor is placed on (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 parameter f_{ijp}. $f_{ijp} = 1$ if there is 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 contaminated water (such as treatment cost) and converts risk into financial terms.

In the objective, 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. Uncertain flow demands of known probability distributions at various nodes result in multiple flow patters p in the network. It is necessary to consider all these patterns to simulate every possibility of contamination propagation. This makes the problem stochastic. The uncertain space is characterized here through N_{samp} samples in order to use sampling based L-shaped BONUS algorithm.

The first stage problem uses s_{ij} as the decision variables to minimize total sensor cost and expected risk which constitutes the recourse part of the problem. The constraint in the first stage problem restricts the maximum number of sensors used. Second stage problem minimizes the expected risk for N_{samp} realizations of uncertain demands. c_{ipj} constitute the second stage decision variables. For this second stage problem, first set of constraints ensures that when a node is directly attacked, it is contaminated while the second set propagates 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 [32, 33, 34] for details about the problem formulation.

The network selected for the case study is a modification of the “Example Network 1” from EPANET [35, 31]. The network is shown in figure 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 which give 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. Yet, from practical point of view, this constitutes a trivial problem in that a realistic water distribution network will have hundreds of nodes and more branches. The simulation of such networks can be a cumbersome task.

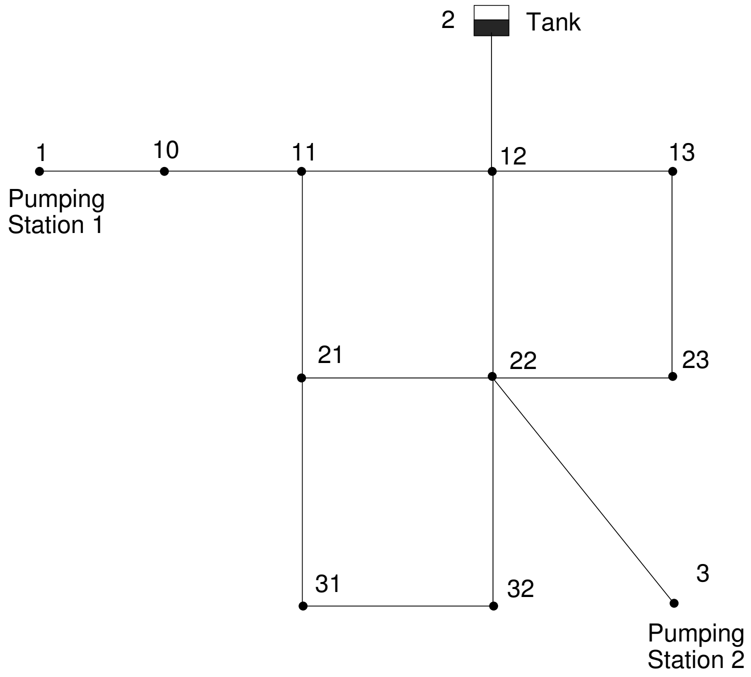

Figure 11: Water distribution network for the sensor placement application of L-shaped BONUS

8.2 Verification of the reweighting scheme

Before the problem solution and results are reported, the validity of using 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 samples size is shown in figure 12. The accuracy of estimation increases with increase in the sample size as the relative error between the estimated and actual results goes down, as shown in figure 12.

8.3 Problem solution

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

In the standard L-shaped with sampling method, 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 will be computationally very inconvenient, as it needs linking of the optimization code with the EPANET simulation software and running the simulation for each sample.

The proposed algorithm instead simplifies the task by following the path shown in figure 5. So the EPANET is first simulated for a certain number of samples to get the distribution of 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, reweighting scheme was used to estimate the distribution of various flow patterns. This algorithm does not need to connect the optimization code with the EPANET simulator as the first simulations can be done off-line and the results stored to be 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 actual objective function and 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 stochastic simulation results in decision making, the estimated and actual values for this method are same.

A comparison of results shows that the estimated values for the two methods are differ-

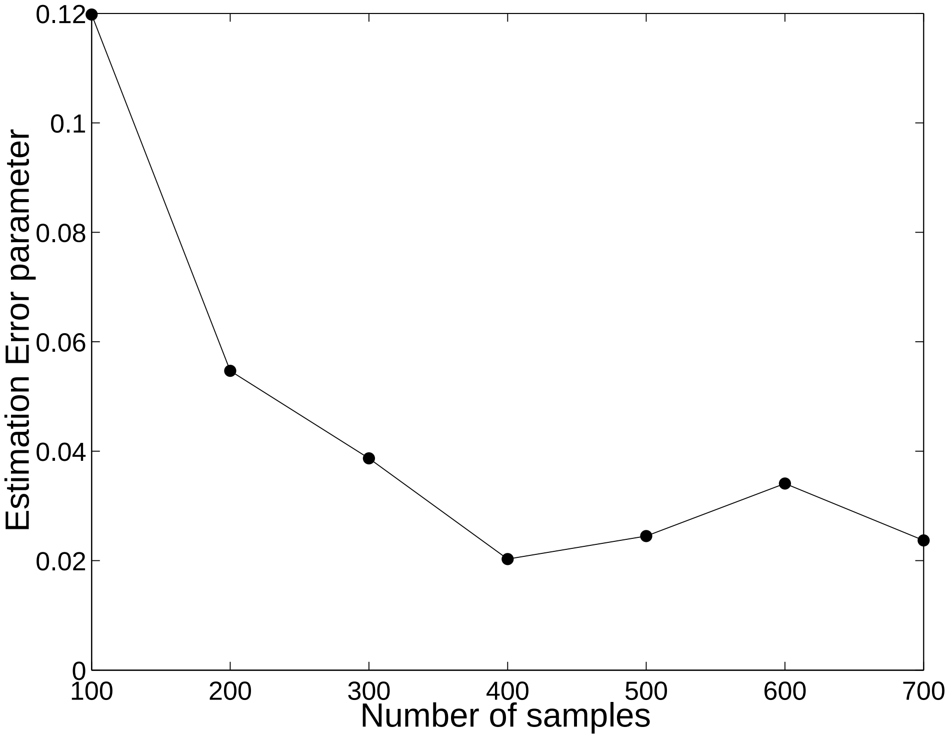

Figure 12: Estimation error dependence on number of samples

ent, those for deterministic being lower than stochastic method. The actual objective function and percentage risk values are however more important for comparing 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 sub-optimal and 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 objective function values in table 4 as well as the placement locations of 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 solution possible with considerable ease. This is the important result from this case study.

9 Conclusion

The paper proposes a new algorithm to solve stochastic nonlinear programming problems. 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, also to solve SNLP problems. 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 SNLP problems with considerably reduced computational burden. In all the cases, reweighting approximation is shown not to compromise accuracy severely while greatly reducing the computation times. It was also shown for the first two problems that HSS technique of sampling is better than 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 which 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 additional properties and application areas.

Acknowledgements

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

References

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

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

[3] J-H Rya, V. Dua, and E.N. Pistikopoulos. A bilevel programming framework for enterprise-wide process networks under uncertainty. *Computers and Chemical Engineering*, 28:1121–1129, 2004.

Table 4: Comparative results of different solution methods

Maximum number of allowed sensors	Type of sensor	Deterministic analysis				Stochastic analysis			
		Estimated cost ($) $(\times 10^7)$	Estimated percentage risk	Actual cost ($) $(\times 10^7)$	Actual percentage risk	Estimated cost ($) $(\times 10^7)$	Estimated percentage risk	Actual cost ($) $(\times 10^7)$	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

[4] J.Y. Jung, G. Blau, J.F. Penky, G.V. Reklaitis, and D. Eversdyk. A simulation based optimization approach to supply chain management under demand uncertainty. *Computers and Chemical Engineering*, 28:2087–2106, 2004.

[5] X. Lin, S.L. Janak, and C.A. Floudas. A new robust optimization approach for scheduling under uncertainty: I. bounded uncertainty. *Computers and Chemical Engineering*, 28:1069–1085, 2004.

[6] U. Diwekar. Greener by design. *Environmental Science and Technology*, 37:5432–5444, 2003.

[7] U. Diwekar. Green process design, industrial ecology, and sustainability: A systems analysis perspective. *Resources, conservation and recycling*, 44(3):215–235, 2005.

[8] S. Kheawhom and M. Hirao. Decision support tools for environmentally benign process design under uncertainty. *Computers and Chemical Engineering*, 28:1715–1723, 2004.

[9] A. Charnes and W.W. Cooper. Chance-constrained programming. *Management Science*, 5:73–79, 1959.

[10] W.T. Ziemba. Solving nonlinear programming problems with stochastic objective functions. *The journal of financial and quantitative analysis*, 7(3):1809–1827, Jun. 1972.

[11] S.E. Elmaghraby. Allocation under uncertainty when the demand has continuous d.f. *Management Science*, 10:270–294, 1960.

[12] W.T. Ziemba. Computational algorithms for convex stochastic programs with simple recourse. *Operations Research*, 18:414–431, Jun. 1970.

[13] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer series in Operations Research, 1997.

[14] A. Ruszczyński. A regularized decomposition for minimizing a sum of polyhedral functions. *Mathematical programming*, 35:309–333, 1986.

[15] F.V. Louveaux. Piecewise convex programs. *Mathematical programming*, 15:53–62, 1978.

[16] R.T. Rockafellar and R.J-B Wets. A lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming. *Mathematical Programming Study*, 28:63–93, 1986.

[17] R.T. Rockafellar and R.J-B Wets. Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[18] F.V. Louveaux. Nonlinear stochastic programming. Technical report, Faculté des Sciences, Département de Mathématique, 2001.

[19] G.B. Dantzig and P.W. Glynn. Parallel processors for planning under uncertainty. *Annals of Operations Research*, 22:1–21, 1990.

[20] G.B. Dantzig and G. Infanger. *Computational and Applied Mathematics I*. Elsevier Science Publishers, B.V. (North Holland), 1992.

[21] J. Higle and S. Sen. Stochastic decomposition: an algorithm for two stage linear programs with recourse. *Annals of Operations Research*, 16:650–669, 1991.

[22] K.H. Sahin and U.M. Diwekar. 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, 2004.

[23] T. Hesterberg. Weighted average importance sampling and defensive mixture distribution. *Technometrics*, 37:185–194, 1995.

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

[25] R. Van Slyke and R.J-B Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[26] U.M. Diwekar. *Introduction to Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2003.

[27] J.R. Kalagnanam and U. Diwekar. An efficient sampling technique for off-line quality control. *Technometrics*, 39(3):308–319, 1997.

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

[29] Forest Gray. *Petroleum production for the nontechnical person*. Tulsa, Okla.: PennWell Books, 1986.

[30] A.K. Rhodes. Refinery operating variables key to enhanced lube oil quality. *Oil and Gas Journal*, January:45, 1993.

[31] L.A. Rossman. *EPANET users manual*. Risk reduction engg. lab, Environmental Protection Agency, Cincinnati, Ohio, 1993.

[32] Y. Shastri and U. Diwekar. Sensor placement in water networks: A stochastic programming approach. *Journal of Water Resources Planning and Management (Accepted for publication)*, 2004.

[33] J. Berry, L. Fleischer, W. Hart, C. Phillips, and J. Watson. Sensor placement in municipal water networks. *To appear in special issue of Journal of Water Res. Planning and Management*, 2005.

[34] A. Kessler, A. Ostfeld, and G. Sinai. Detecting accidental contaminations in municipal water networks. *Journal of Water Res. Planning and Management*, 124:192–198, 1998.

[35] L.A. Rossman. The epanet programmer's toolkit for analysis of water distribution systems. In *Proceedings of the Annual Water Resource Planning and Management Conference*, 1999.