SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

Volume 13 Number 1 March 2002

Contents

Optimization under Uncertainty

Optimization Under Uncertainty: An Overview

Urmila Diwekar

CUSTOM (Center for Uncertain Systems: Tools for Optimization and Uncertainty), Carnegie Mellon University, Pittsburgh, PA 15213 (urmila@cmu.edu).[1]

[1]This overview is based on the chapter entitled "Optimization Under Uncertainty" from *Introduction to Applied Optimization*[1].

1. Introduction

Deterministic optimization literature classifies problems as Linear Programming (LP), NonLinear Programming (NLP), Integer Programming (IP), Mixed Integer LP (MILP), and Mixed Integer NLP (MINLP), depending on the decision variables, objectives, and constraints. However, the future cannot be perfectly forecasted but instead should be considered random or uncertain. Optimization under uncertainty refers to this branch of optimization where there are uncertainties involved in the data or the model, and is popularly known as Stochastic Programming or stochastic optimization problems. In this terminology, stochastic refers to the randomness, and programming refers to the mathematical programming techniques like LP, NLP, IP, MILP, and MINLP. In discrete (integer) optimization, there are probabilistic techniques like Simulated Annealing and Genetic Algorithms; these techniques are sometimes referred to as the stochastic optimization techniques because of the probabilistic nature of the method. In general, however, Stochastic Programming and stochastic optimization involves optimal decision making under uncertainties.

2. Types of Problems and Generalized Representation

The need for including uncertainty in complex decision models arose early in the history of mathematical programming. The first model forms, involving action followed by observation and reaction (or recourse), appear in [2, 3]. The commonly used example of a recourse problem is the news vendor or the newsboy problem described below. This problem has a rich history that has been traced back to the economist Edgeworth [4], who applied a variance

to a bank cash-flow problem. However, it was not until the 1950s that this problem, like many other OR/MS models seeded by the war effort, became a topic of serious and extensive study by academicians [5].

Example: The simplest form of a stochastic program may be the news vendor (also known as the newsboy) problem. In the news vendor problem, the vendor must determine how many papers (x) to buy now at the cost of c cents for a demand which is uncertain. The selling price is s_p cents per paper. For a specific problem, whose weekly demand is shown below, the cost of each paper is $c = 20$ cents and the selling price is $s_p = 25$ cents. Solve the problem, if the news vendor knows the demand uncertainties but does not know the demand curve for the coming week a-priori (Table 1). Assume no salvage value $s = 0$, so that any papers bought in excess of demand are simply discarded with no return.

Table 1: Weekly demand and its uncertainties.

Weekly demand			Uncertainty		
i	Day	Demand (d_i)	j	Demand (d_j)	Probability (p_j)
1	Monday	50	1	50	5/7
2	Tuesday	50	2	100	1/7
3	Wednesday	50	3	140	1/7
4	Thursday	50			
5	Friday	50			
6	Saturday	100			
7	Sunday	140			

Solution: In this problem, we want to find how many papers the vendor must buy (x) to maximize the profit. Let r be the effective sales and w be the excess which is going to be thrown away. This problem falls under the category of Stochastic Programming with recourse where there is action (x), followed by observation (profit), and reaction (or recourse) (r and w). We know that any papers bought in the excess are just thrown away. It is obvious that we should minimize the waste but increase the sales. Our first instinct to solve this problem is to find the average demand and find the optimal supply x corresponding this demand. Since the average demand from the Table 1 is 70 papers, $x = 70$ should be the solution. Let us see if this represents the optimal solution for the problem. Table 2 shows the observation (profit function) for this action.

Table 2: Supply and profit.

i	Day	Supply, x_i	Profit, cents
1	Monday	70	-150
2	Tuesday	70	-150
3	Wednesday	70	-150
4	Thursday	70	-150
5	Friday	70	-150
6	Saturday	70	350
7	Sunday	70	350
Average Weekly		-	- 50

From Table 2, it is obvious that if we take the average demand as the solution, then the news vendor will be making a loss of 50 cents per week. This probably is not the optimal solution. Can we do better? For that we need to propagate the uncertainty in the demand to see the effect of uncertainty on the objective function and then find the optimum value of x. This formulation is shown below.

$$\underset{x}{\text{Maximize}} \quad Z = \text{Profit}_{\text{avg}}(u)$$

subject to

$$\begin{aligned} \text{Profit}_{\text{avg}}(u) &= \int_0^1 [-cx + \text{Sales}(r, w, p(u))]dp \\ &= \sum_j p_j \,\text{Sales}(r, w, d_j) - cx \\ \text{Sales}(r, w, d_j) &= s_p\, r_j \;+\; s w_j \end{aligned}$$

where

$$\begin{aligned} r_j &= \min(x, d_j) \\ &= x, \quad \text{if} \quad x \le d_j \\ &= d_j, \quad \text{if} \quad x \ge d_j \\ w_j &= \max(x - d_j, 0) \\ &= 0, \quad \text{if} \quad x_i \le d_i \\ &= x_i - d_i, \quad \text{if} \quad x_i \ge d_i \end{aligned}$$

The above information can be transformed for daily profit as follows: if $d_1 \le x \le d_2$,

$$\text{Profit} = -cx + 5/7 s_p\, d_1 + 1/7 s_p\, x + 1/7 s_p\, x, \quad (1)$$

or if $d_2 \le x \le d_3$,

$$\text{Profit} = -cx + 5/7 s_p\, d_1 + 1/7 s_p\, d_2 + 1/7 s_p\, x. \quad (2)$$

Notice that the problem represents two equations for the objective function, Equations 1 and 2, making the objective function a discontinuous function and is no longer an LP. Special methods like the L-shaped decomposition or stochastic decomposition are required to solve this problem. However, since the problem is simple, we can solve this problem as two separate LPs. The two possible solutions to the above LPs are $x = d_1 = 50$ and $x = d_2 = 100$ respectively. This provides the news vendor with an optimum profit of 1750 cents per week from Equation 1 and with a loss of 2750 cents per week from Equation 2. Obviously Equation 1 provides the global optimum for this problem.

The difference between taking the average value of the uncertain variable as the solution as compared to using stochastic analysis (propagating the uncertainties through the model and finding the effect on the objective function as above) is defined as the *Value of Stochastic Solution, VSS*. If we take the average value of the demand, i.e. $x = 70$, as the solution, we obtain a loss of 50 cents per week. Therefore, the value of stochastic solution, VSS, is $1750 - (-50) = 1800$ cents per week.

Now consider the case where the vendor knows the exact demand (Table 1) a-priori. This is the perfect information problem where we want to find the solution x_i for each day i. Let us formulate the problem in terms of x_i.

$$\underset{x_i}{\text{Maximize}} \quad \text{Profit}_i = -cx_i + \text{Sales}(r, w, d_i)$$

subject to

$$\begin{aligned} \text{Sales}(r, w, d_i) &= s_p\, r_i + s w_i \\ r_i &= \min(x_i, d_i) \\ &= x_i, \quad \text{if} \quad x_i \le d_i \\ &= d_i, \quad \text{if} \quad x_i \ge d_i \\ w_i &= \max(x_i - d_i, 0) \\ &= 0, \quad \text{if} \quad x_i \le d_i \\ &= x_i - d_i, \quad \text{if} \quad x_i \ge d_i \end{aligned}$$

Here we need to solve each problem (for each i) separately, leading to the following decisions shown in Table 3.

Table 3: Supply and profit.

i	Day	Supply, x_i	Profit, cents
1	Monday	50	250
2	Tuesday	50	250
3	Wednesday	50	250
4	Thursday	50	250
5	Friday	50	250
6	Saturday	100	500
7	Sunday	140	700
Average Weekly		-	2450

One can see that the difference between the two values, (1) when the news vendor has the perfect information and (2) when he does not have the perfect information but can represent it using probabilistic functions, is the Expected Value of Perfect Information, EVPI. EVPI is 700 cents per week for this problem.

The literature on optimization under uncertainties very often divides the problems into categories such as "wait and see", "here and now" and "chance constrained optimization" [6, 7]. In "wait and see" we wait until an observation is made on the random elements, and then solve the deterministic problem. The last formulation described in terms of problem under perfect information, falls under this category. This is similar to the "wait and see" problem of Madansky [8], originally called "Stochastic Programming" by Tintner [9], is not in a sense, one of decision analysis. In decision making, the decisions have to be made "here and now" about the activity levels. The "here and now" problem involves optimization over some probabilistic measure— usually the expected value. By this definition, chance constrained optimization problems can be included in this particular category of optimization under uncertainty. Chance constrained optimization involves constraints which are not expected to be always satisfied; only in a proportion of cases, or "with given probabilities".

These various categories require different methods for obtaining their solutions. As stated earlier, we can easily divide these problems into two main categories (1) here and now problems, and the (2) wait and see problems. It should be noted that many

problems have both here and now, and wait and see problems embedded in them. The trick is to divide the decisions into these two categories and use a coupled approach.

Here and Now Problems: The "here and now" problems require that the objective function and constraints be expressed in terms of some probabilistic representation (e.g. expected value, variance, fractiles, most likely values). For example, in chance constrained programming, the objective function is expressed in terms of expected value, while the constraints are expressed in terms of fractiles (probability of constraint violation), and in the Taguchi's off-line quality control method [10], the objective is to minimize variance. These problems can be classified as here and now problems.

The "here and now" problem, where the decision variables and uncertain parameters are separated, can then be viewed as:

$$\text{Optimize} \;\; J \;=\; P_1(j(x,u)) \tag{3}$$

subject to

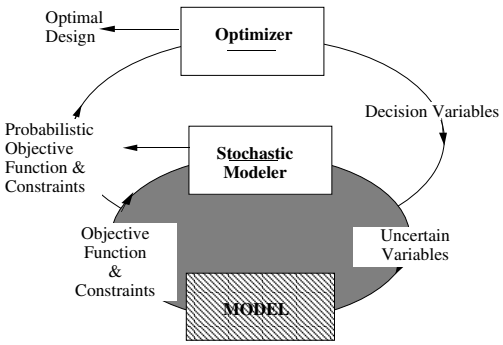
$$P_2(h(x,u)) = 0$$
$$P_3(g(x,u) \geq 0) \geq \alpha$$

where u is the vector of uncertain parameters and P represents the cumulative distribution functional such as the expected value, mode, variance, or fractiles.

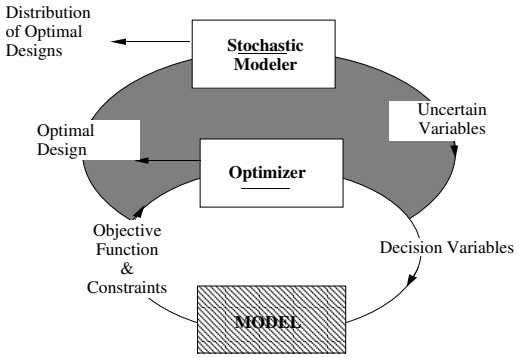
Unlike the deterministic optimization problem, in stochastic optimization one has to consider the probabilistic functional of the objective function and constraints. The generalized treatment of such problems is to use probabilistic or stochastic models instead of the deterministic model inside the optimization loop.

Figure 1a represents the generalized solution procedure, where the deterministic model is replaced by an iterative stochastic model with sampling loop representing the discretized uncertainty space. The uncertainty space is represented in terms of the moments like the mean, fractiles, or the standard deviation of the output. In chance constrained formulation, the uncertainty surface is translated into input moments, resulting in an equivalent deterministic optimization problem. This is discussed in the next section.

Wait and See: In contrast to here and now problems, which yield optimal solutions that achieve a given level of confidence, wait and see problems involve a category of formulations that shows the effects of uncertainty on optimum design. A wait and see problem involves deterministic optimal decision at each scenario or random sample, equivalent to solving several deterministic optimization problems. The generalized representation of this problem is given below.



(a) Here and Now



(b) Wait and See

Figure 1: Pictorial representation of the stochastic programming framework.

$$\text{Optimize} \;\; Z = z(x,u*) \tag{4}$$

subject to

$$h(x,u*) = 0$$
$$g(x,u*) \leq 0$$

where $u*$ is the vector of values of uncertain variables corresponding to each scenario or sample. This optimization procedure is repeated for each sample of uncertain variables u and a probabilistic representation of the outcome is obtained.

Figure 1b represents the generalized solution procedure, where the deterministic problem forms the inner loop, and the stochastic modeling forms the outer loop. The difference between the two solutions obtained using the two frameworks is the Expected Value of Perfect Information (EVPI). The concept of EVPI was first developed in the context of decision analysis and can be found in classical references such as [11]. From Figures 1 it is clear that by simply interchanging the position of the uncertainty analysis framework and the optimization framework, one can solve many problems in stochastic optimization and Stochastic Programming domain [1]. Recourse problems with multiple stages involve decisions that are taken before the uncertainty realization (here and now) and recourse actions which can be taken when information is disclosed (wait and see). These problems can be solved using decomposition methods.

As can be seen from the above description, both here and now and wait and see problems require representation of uncertainties in the probabilistic space and then the propagation of these uncertainties through the model to obtain the probabilistic representation of the output. This is the major difference between stochastic and deterministic optimization problems. Is it possible to propagate the uncertainty using moments (like mean and variance), thereby obtaining a deterministic representation of the problem? This is the basis of the chance constrained programming method, developed very early in the history of optimization under uncertainty, principally by Charnes and Cooper [12].

3. Chance Constrained Programming Method

In the chance constrained programming (CCP) method, some of the constraints likely need not hold as we had assumed in earlier problems. Chance constrained problems can be represented as follows:

$$\text{Optimize} \quad J = P_1(j(x,u)) = E(z(x,u)) \tag{5}$$

subject to

$$P(g(x) \leq u) \ \leq \ \alpha \tag{6}$$

In the above formulation, equation 6 is the chance constraint. In chance constraint formulation, this constraint (or constraints) is (are) converted into a deterministic equivalent under the assumption that the distribution of the uncertain variables, u, is a stable distribution. Stable distributions are such that the convolution of two distribution functions $F(x - m_1/v_1)$ and $F(x - m_2/v_2)$ is of the form $F(x - dmu/v)$, where m_i and v_i are two parameters of the distribution [13]. Normal, Cauchy, Uniform, and Chi-square are all stable distributions that allow the conversion of probabilistic constraints into deterministic ones. The deterministic constraints are in terms of moments of the uncertain variable u (input uncertainties). For example, if the constraint g in equation 6 has a cumulative probability distribution F then the deterministic equivalent of this constraint is given below. Deterministic equivalent of the chance constraint 6:

$$g(x) \ \leq \ F^{-1}(\alpha) \tag{7}$$

where F^{-1} is the inverse of the cumulative distribution function F.

The major restrictions in applying the CCP formulation include that the uncertainty distributions should be stable distribution functions, the uncertain variables should appear in the linear terms in the chance constraint, and that the problem needs to satisfy the general convexity conditions. The advantage of the method is that one can apply the deterministic optimization techniques to solve the problem.

4. Uncertainty Analysis and Sampling

The probabilistic or stochastic modeling (Figure 1) iterative procedure involves:

1. Specifying the uncertainties in key input parameters in terms of probability distributions.
2. Sampling the distribution of the specified parameter in an iterative fashion.
3. Propagating the effects of uncertainties through the model and applying statistical techniques to analyze the results.

4.1 Specifying Uncertainty Using Probability Distributions

To accommodate the diverse nature of uncertainty, the different distributions can be used. The type

of distribution chosen for an uncertain variable reflects the amount of information that is available. For example, the uniform and log-uniform distributions represent an equal likelihood of a value lying anywhere within a specified range, on either a linear or logarithmic scale, respectively. Further, a normal (Gaussian) distribution reflects a symmetric but varying probability of a parameter value being above or below the mean value. In contrast, lognormal and some triangular distributions are skewed such that there is a higher probability of values lying on one side of the median than the other. Once probability distributions are assigned to the uncertain parameters, the next step is to perform a sampling operation from the multi-variable uncertain parameter domain.

4.2 Sampling Techniques

One of the most widely used techniques for sampling from a probability distribution is the Monte Carlo sampling technique, which is based on a pseudo-random generator used to approximate a uniform distribution (i.e., having equal probability in the range from 0 to 1). The specific values for each input variable are selected by inverse transformation over the cumulative probability distribution.

Importance Sampling: Crude Monte Carlo methods can result in large error bounds (confidence intervals) and variance. Variance reduction techniques are statistical procedures designed to reduce the variance in the Monte Carlo estimates [14]. Importance sampling, Latin Hypercube Sampling (LHS) [15, 16], Descriptive Sampling [17], and Hammersley Sequence Sampling [18] are examples of variance reduction technique. In importance Monte Carlo sampling, the goal is to replace a sample using the distribution of u with one that uses an alternative distribution that places more weight in the areas of importance. Obviously such a distribution function is problem dependent and is difficult to find. The following two sampling methods provide a generalized approach to improve the computational efficiency of sampling.

Latin Hypercube Sampling: The main advantage of Monte Carlo method lies in the fact that the results from any Monte Carlo simulation can be treated using classical statistical methods; thus results can be presented in the form of histograms, and methods of statistical estimation and inference are applicable. Nevertheless, in most applications, the actual relationship between successive points in a sample has no physical significance; hence the randomness/independence for approximating a uniform distribution is not critical [19]. Once it is apparent that the uniformity properties are central to the design of sampling techniques, constrained or stratified sampling techniques become appealing [20].

Latin hypercube sampling [15, 16] is one form of stratified sampling that can yield more precise estimates of the distribution function. In Latin hypercube sampling, the range of each uncertain parameter X_i is sub-divided into non-overlapping intervals of equal probability. One value from each interval is selected at random with respect to the probability distribution in the interval. The n values thus obtained for X_1 are paired in a random manner (i.e., equally likely combinations) with n values of X_2. These n values are then combined with n values of X_3 to form n-triplets, and so on, until n k-tuplets are formed. In median Latin Hypercube sampling (MLHS) this value is chosen as the mid-point of the interval. MLHS is similar to the descriptive sampling described by [17]. The main drawback of this stratification scheme is that it is uniform in one dimension and does not provide uniformity properties in k-dimensions.

Hammersley Sequence Sampling: Recently, an efficient sampling technique (Hammersley sequence sampling) based on Hammersley points has been developed by Kalagnanam and Diwekar[18], which uses an optimal design scheme for placing the n points on a k-dimensional hypercube. This scheme ensures that the sample set is more representative of the population, showing uniformity properties in multi-dimensions, unlike Monte Carlo, Latin hypercube, and its variant, the Median Latin hypercube sampling techniques. The paper by Kalagnanam and Diwekar[18] provides a comparison of the performance of the Hammersley sampling (HSS) technique to that of other techniques. It was found that the HSS technique is at least 3 to 100 times faster than LHS and Monte Carlo techniques and hence is a preferred technique for uncertainty analysis as well as optimization under uncertainty.

4.3 Sampling Accuracy and Different Algorithms

As stated earlier, the Stochastic Programming formulations often include some approximations of the underlying probability distribution. The disadvantage of sampling approaches that solve the γ-th approximation completely is that some effort might be wasted on optimizing when approximation is not accurate [21]. For specific structures where the L-shaped method is applicable, two approaches avoid these problems by embedding sampling within another algorithm without complete optimization. These two approaches are the method of Dantzig[22] which uses importance sampling to reduce variance in each cut based on a large sample, and the stochastic decomposition method proposed by Higle and Sen[23]. Please refer to article by Dr. Rico-Ramirez on stochastic linear programming for details.

In almost all stochastic optimization problems, the major bottleneck is the computational time involved in generating and evaluating probabilistic functions of the objective function and constraints. The accuracy of the estimates for the actual mean (μ) and the actual standard deviation (σ) is particularly important to obtain realistic estimates of any performance or economic parameter. However, this accuracy is dependent on the number of samples. The number of samples required for a given accuracy in a stochastic optimization problem depends upon several factors, such as the type of uncertainty, and the point values of the decision variables[24]. Especially for optimization problems, the number of samples required also depends on the location of the trial point solution in the optimization space. Figure 2 shows how the shape of the surface over a range of uncertain parameter values changes since one is at a different iteration (different values of decision variables) in the optimization loop. Therefore, the selection of the number of samples for the stochastic optimization procedure is a crucial and challenging problem. A combinatorial optimization algorithm that automatically selects the number of samples and provides the trade-off between accuracy and efficiency is presented in the article by Dr. Ki-Joo Kim in this issue.

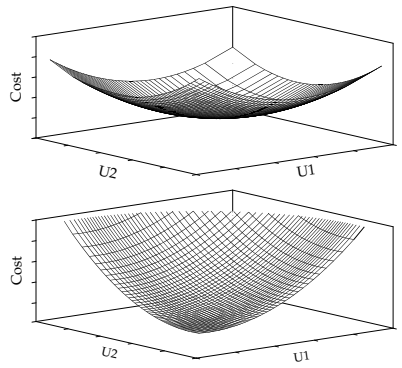


Figure 2: Uncertainty space at different optimization iterations.

5. Summary

The problems in optimization under uncertainty involve probabilistic objective function and constraints. These problems can be categorized as (1) here and now problems, and (2) the wait and see problems. Many problems involve both here and now, and wait and see decisions. The difference in the solution of these two formulations is the expected value of perfect information (EVPI). Recourse problems normally involve both here and now, and wait and see decisions and hence are normally solved by decomposition strategies like the L-shaped method. The major bottleneck in solving stochastic optimization (programming) problems is the propagation of uncertainties. In chance constrained programming, the uncertainties are propagated as moments, resulting in a deterministic equivalent problem. However, chance constrained programming methods are applicable to a limited number of problems. A generalized approach to uncertainty propagation involves sampling methods that are computationally intensive. New sampling techniques like the Hammersley Sequence sampling reduce the computational intensity of the sampling approach. Sampling error bounds can be used to reduce the computational intensity of the stochastic optimization procedure further. This strategy is used in some of the decomposition methods and in the stochastic annealing algorithm described in the next two articles. For stochastic nonlinear problems, a new algorithm called Better Optimization of Nonlinear Uncertain Systems (BONUS) is also presented in the last article by Dr. Kemal Sahin.

REFERENCES

[1] Diwekar U. M., *Introduction to Applied Optimization*, to be published by Kluwer Academic Publishers, Boston, MA, 2002.

[2] Bealet E.M. L., *On minimizing a convex function subject to linear inequalities*, J. Royal Statistical Society, 17B: 173.

[3] Dantzig G. B., *Linear programming under uncertainty*, Management Science, 1: 197.

[4] Edgeworth E.,*The mathematical theoryof banking*, J. Royal Statistical Society, 1888, 51: 113.

[5] Petruzzi N. C. and M. Dada, *Pricing and the newsvendor problem: a review with extensions*, Operations Research, 1999, 47(2): 183.

[6] Vajda S., *Probabilistic programming*, Academic Press, New York, 1972.

[7] Nemhauser, G. L., A. H. G. Ronnooy Kan, and M. J. Todd, *Optimization: Handbooks in operations research and management science*, Vol. 1. North-Holland Press: New York, 1989.

[8] Madansky A., *Inequalities for stochastic linear programming problems*, Man. Sci., 1960, 6: 197.

[9] Tintner G., *Stochastic linear programming with applications to agricultural economics*, *In Proc. 2nd Symp. Lin. Progr.* , Washington, 1955, 197.

[10] Taguchi G., *Introduction to Quality Engineering*, Asian Productivity Center, Tokyo, Japan, 1986.

[11] Raiffa H. and R. Schlaifer, *Applied Stastical Decision Theory*, Harvard University, Boston, MA, 1961.

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

[13] Luckacs, *Characteristic Functions*, Charles Griffin, London, 1960.

[14] James B. A. P., *Variance reduction techniques*, J. Operations Research Society, 1985, 36(6): 525.

[15] McKay M. D., R. J. Beckman, and W. J. Conover, *A comparison of three methods of selecting values of input variables in the analysis of output from a computer code*, Technometrics, 1979, 21(2): 239.

[16] Iman R. L. and W. J. Conover, *Small sample sensitivity analysis techniques for computer models, with an application to risk assessment*, Communications in Statistics, 1982, A17: 1749.

[17] Saliby E., *Descriptive sampling: A better approach to Monte Carlo simulations*, J. Operations Research Society, 1990, 41, no.12, 1133.

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

[19] Knuth D. E.,*The Art of Computer Programming, Volume 1: Fundamental Algorithms*, Addison-Wesley Publishing Co., 1973.

[20] Morgan G. and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, UK, 1990.

[21] Birge, J. R. and F. Louveaux, *Introduction to stochastic programming*, Springer-Verlag, New York, 1997.

[22] Dantzig G. B. and G. Infanger, *Large scale stochastic linear programs–Importance sampling and bender decomposition*, Computational and Applied Mathematics, Brezinski and U. Kulisch Eds., 1991, 111.

[23] Higle, J. L. and S. Sen, *Stochastic decomposition: an algorithm for two-stage linear programs with recourse*, Mathematics of Operations Research, 1991, 16(3): 650-669.

[24] Painton, L. A. and U. M. Diwekar, *Stochastic annealing under uncertainty*, European Journal of Operations Research, 1995, 83: 489.

Two-Stage Stochastic Linear Programming: A Tutorial

Vicente Rico-Ramirez

Instituto Tecnologico de Celaya, Mexico

(vicente@iqcelaya.itc.mx)[1]

1. Introduction

There is a huge body of literature on stochastic linear programming including surveys and numerous articles (e.g., [2, 3, 6]). The main class of stochastic linear problems with recourse (*SLPwR*) involves two stages. In the first stage, the choice of the decision variable x is made. In the second stage, following the observation of the value of u and the evaluation of the objective function, a corrective action (represented by the second stage decisions, v) is suggested.

The standard mathematical form of a *SLPwR* problem is given by Equations (1) and (2). Equation (1) is referred as the first stage problem :

$$\begin{array}{lll} \text{Min} & c^T x + Q(x) & \\ s.\,t. & Ax \;\alpha\; b & \\ & x \geq 0 & \end{array} \tag{1}$$

where A is a coefficient matrix, c is a coefficient vector, α is used to represent any of $=, >$ or $<$, $Q(x)$ is the recourse function defined by $Q(x) = E_u[Q(x,u)]$, and $Q(x,u)$ is obtained from the second stage problem, Equation (2):

$$\begin{array}{lll} Q(x,u) = & \text{Min} & q^T(u)\; v \\ & s.\,t. & W(u)\; v \;\;\alpha\;\; h(u) - T(u)\; x \end{array} \tag{2}$$

[1]Visiting scientist, CUSTOM, Carnegie Mellon University.

$$v \geq 0$$

where q is a coefficient vector and W, h and T are coefficient matrices which in principle might depend on the random variables u. The matrix W is of particular interest and is known as the recourse matrix.

Observe that the problem given by Equation (1) has linear constraints and a convex objective function, so that there is a rather complete optimization theory with necessary and sufficient conditions. However, *SLPwR* problems are difficult to solve since in general it is very demanding to generate evaluations of the recourse function ($Q(x)$) and its gradient. Some of the difficulties disappear when the recourse structure is simple, as in problems with fixed and complete recourse. Fixed recourse means that the recourse matrix, W, is independent on u (simply a coefficient matrix), whereas complete recourse means that any set of values that we choose for the first stage decisions, x, leaves us with a feasible second stage problem.

The two main algorithms for stochastic linear programming with fixed recourse are the L-Shaped method [1, 2] and the Stochastic Decomposition algorithm (SD) [4, 5]:

1. The L-Shaped method is used when the random variables of the problem are described by **discrete** distribution functions. As a result, **exact** computations for the lower bound of the recourse function are possible.
2. On the other hand, the SD algorithm uses **sampling** when the random variables are represented by continuous **distribution** functions. As a result, **estimations** for the lower bound of the recourse function are based on expectation.

2. Feasibility and Optimality Cuts

Both of the algorithms (L-Shaped and SD) are based on the addition of linear constraints (known as cuts) to the first stage problem. Hence, there are two types of cuts successively added during the solution procedure: Feasibility cuts and optimality cuts.

A feasibility cut is a linear constraint which ensures that a first stage decision is second stage feasible. Notice that a complete recourse problem does not need the addition of feasibility cuts. There exists a formal notation to indicate that any decision we take on the first stage should leave us with a second stage problem which is feasible:

$$\begin{array}{ll} \text{Min} & c^T x + Q(x) \\ s.\,t. & x \in (K_1 \cap K_2) \end{array} \tag{3}$$

where the set K_1 contains any possible decisions which satisfy the constraints of the first stage, while the set K_2 contains those decisions which are second stage feasible. That is, $K_1 = \{ x |\ Ax = b,\ x \geq 0\}$ and $K_2 = \{ x | Q(x) \leq \infty\}$. Very often, the problem in Equation 3 is reformulated as:

$$\begin{array}{ll} \text{Min} & c^T x + \theta \\ s.\,t. & Q(x) \leq \theta \\ & x \in (K_1 \cap K_2) \end{array} \tag{4}$$

On the other hand, an optimality cut is a linear approximation of $Q(x)$ on its domain of finiteness, and is determined based on the dual of the second stage problem. As such, each optimality cut provides a lower bound (linear support) on $Q(x)$. The dual of the second stage problem defined by Equation (2) is given by:

$$\begin{array}{ll} \text{Max} & \pi^T\ (h\ -\ Tx) \\ s.\,t. & \pi^T\ W \leq q \end{array} \tag{5}$$

where the dual variables, π, corresponds also to the lagrange multipliers of the problem given by Equation (2). Next we show how these types of cuts are derived ([2]).

2.1 Optimality Cut

Assume that we are solving a *SLPwR* with either the L-Shaped or the SD method. Assume also that we are performing the $\nu - th$ iteration of the algorithm which considers the $k - th$ realization of the random variables. Using the duality theorem, the value of the objective function of the second stage problem would be given by:

$$Q(x^\nu, u^k)\ =\ (\pi_k^\nu)^T\ (h_k\ -\ T_k\ x^\nu) \tag{6}$$

and therefore, because of convexity:

$$Q(x, u^k) \geq (\pi_k^\nu)^T\ (h_k\ -\ T_k\ x) \tag{7}$$

If we consider now a discrete distribution for u (L-Shaped method) and assume that the probability for the $k-th$ realization of u to happen is p_k, then the expected value of the objective function given by Equation (6) is:

$$Q(x^\nu) = E_u[Q(x^\nu, u^k)] = E_u[(\pi_k^\nu)^T(h_k - T_k x^\nu)] \\ = \sum_{k=1}^K p_k \ \left[(\pi_k^\nu)^T \ (h_k \ - \ T_k \ x^\nu))\right] \quad (8)$$

Hence, also because of convexity:

$$Q(x) \geq \sum_{k=1}^K p_k \ \left[(\pi_k^\nu)^T \ (h_k \ - \ T_k \ x)\right] \quad (9)$$

Finally, by defining

$$g = \sum_{k=1}^K p_k (\pi_k^\nu)^T \ h_k, \qquad G = \sum_{k=1}^K p_k (\pi_k^\nu)^T \ T_k$$

and recalling that $\theta \geq Q(x)$, then by substituting g and G in Equation (9), we get the optimality cut as a result:

$$G \ x \ + \ \theta \geq g \quad (10)$$

2.2 Feasibility cut

As explained above, the role of a feasibility cut is to ensure that a first stage decision is second stage feasible. A first stage decision x^ν is second stage feasible if a finite vector v exists such that the constraints of the second stage problem are satisfied. To test for the feasibility of such constraints, one can solve the problem:

$$\begin{array}{rlll} z = & \text{Min} & e^T \ (v^+ \ + \ v^-) & \\ & s.\ t. & W \ v \ + \ v^+ \ - \ v^- \ = \ h \ - T \ x^\nu & (11) \\ & & v \geq 0, v^+ \geq 0, v^- \geq 0 & \end{array}$$

or its equivalent dual:

$$\begin{array}{rlll} z = & \text{Max} & \sigma^T \ (h \ - \ T \ x^\nu) & \\ & s.\ t. & \sigma^T \ W \leq 0 & (12) \\ & & |\sigma| \leq e & \end{array}$$

where e is the vector of ones. Note that the problem given by Equation (11) comes from the modification of the second stage problem by adding the positive variables v^+ and v^-. Clearly, $z \geq 0$. If $z = 0$, it means that the second stage constraints are satisfied and, therefore, x^ν is second stage feasible. However, if $z > 0$, then the original second stage problem is infeasible (and its dual unbounded). Hence, the second stage problem is infeasible because $z = (\sigma^\nu)^T \ (h - \ T \ x^\nu)$ is greater than zero (recall the duality theorem). Therefore, in order to ensure the feasibility of the second stage problem, the constraint:

$$(\sigma^\nu)^T \ (h \ - \ T \ x) \leq 0 \quad (13)$$

must be added. If for some $k-th$ realization of the random variable the second stage problem is infeasible, then, according to Equation (13), we define:

$$D = \ (\sigma^\nu)^T \ T_k, \ \ d = \ (\sigma^\nu)^T \ h_k$$

in order to obtain the feasibility cut:

$$D \ x \geq d \quad (14)$$

3. *SLPwR* Algorithms

In this section the algorithmic steps of the L-Shaped method and the SD method are given.

3.1 L-Shaped Algorithm

The steps of the L-Shaped method are the following ([2]):

- Step 0 Set $r = s = \nu = 0$
- Step 1 Set $\nu = \nu + 1$ and solve the so called Current Problem (CP)

$$\begin{array}{rll} \text{Min} & c^T x + \theta & \\ s.\ t. & Ax \ = \ b & \\ & D_l \ x \geq d_l & l = 1 \ \cdots \ r \\ & G_l \ x \ + \ \theta \geq g_l & l = 1 \ \cdots \ s \end{array}$$

  Let x^ν and θ^ν be the optimal solution. If not optimality cuts exist ($s = 0$), set $\theta^\nu = -\infty$ and do not consider it in CP.

- Step 2 For $k = 1 \ \cdots \ K$ (K is the number of discrete realizations) solve the problem:

$$z = \ \text{Min} \ \ e^T \ (v_k^+ \ + \ v_k^-)$$

$$s.\,t.\quad W\ v_k\ +\ v_k^+\ -\ v_k^-\ =\ h_k\ -T_k\ x^\nu$$
$$v_k \geq 0, v_k^+ \geq 0, v_k^- \geq 0$$

If for any k the optimal value is $z > 0$, add a feasibility cut, set $r = r + 1$ and return to Step 1. Otherwise go to Step 3. The feasibility cut is given by

$$D_{r+1}\ x \geq d_{r+1}$$

where $D_{r+1} = (\sigma_k^\nu)^T\ T_k$, $d_{r+1} = (\sigma_k^\nu)^T\ h_k$, and σ_k^ν are the Lagrange multipliers of the above problem.

- Step 3 For $k = 1\ \cdots\ K$ solve the problem:

$$\begin{array}{lll} \text{Min} & & q_k^T\ v_k \\ s.\,t. & W\ v_k & =\ h_k\ -T_k\ x^\nu \\ & & v_k \geq 0 \end{array}$$

(or its dual) to calculate the Lagrange multipliers π_k^ν and define:

$$g_{s+1}\ =\ \sum_{k=1}^{K} p_k(\pi_k^\nu)^T\ h_k \tag{15}$$

$$G_{s+1}\ =\ \sum_{k=1}^{K} p_k(\pi_k^\nu)^T\ T_k \tag{16}$$

$$u^\nu\ =\ g_{s+1}\ -\ G_{s+1}\ x^\nu$$

If $\theta^\nu \geq u^\nu$, stop, x^ν is the optimal solution. Otherwise, add the optimality cut:

$$\theta^\nu \geq g_{s+1}\ -\ G_{s+1}\ x$$

Set $s = s + 1$, and return to Step 1.

3.2 SD Algorithm

In the SD algorithm, it is necessary to sample at each iteration from a continuous probability distribution. For that reason, the estimation of the lower bound of $Q(x)$ is based on expectation. In the L-Shaped algorithm, the optimality cut is calculated in terms of Equations (15) and (16). On the other hand, in the SD algorithm it is calculated in terms of:

$$g^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^T h_k, \quad G^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^T T_k$$

where ν is the number of samples (which is equal to the number of iterations). The algorithm presented here corresponds to that provided by [5]. Higle and Sen [4, 5] assume complete recourse (no feasibility cuts are needed) and provide a simplification involving a restricted solution set for the dual of the second stage problem in order to decrease the computational effort. The steps of that algorithm are given next.

- Step 0 Set $\nu = 0$, $V_0 = \{\emptyset\}$, $\theta^\nu = -\infty$. x^1 is assumed as given.
- Step 1 Set $\nu = \nu + 1$ and sample to generate an observation u^ν independent of any previous observation.
- Step 2 Determine the coefficients of a piecewise linear approximation to $Q(x)$:

  a) Solve the program

$$\begin{array}{lll} \text{Max} & \pi^T\ (h_\nu\ -\ T_\nu\ x^\nu) \\ s.\,t. & \pi^T\ W \leq q \end{array}$$

  to find the values of the vector π, π_ν^ν, and make $V_\nu = V_{\nu-1} \cup \pi_\nu^\nu$

  b) Get the coefficients of the optimality cut

$$g_\nu^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^T h_k, \quad G_\nu^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^T T_k$$

  where π_k^ν is the solution to the problem (for all $k|k \neq \nu$):

$$\begin{array}{lll} \text{Max} & \pi^T\ (h_k\ -\ T_k\ x^\nu) \\ s.\,t. & \pi\ \in\ V_\nu \end{array}$$

  Observe that the solution vector to this problems can only be one of the vectors already included in the set of solutions, V_ν (the solution set is restricted to decrease effort).

  c) Update the coefficients of previous cuts

$$\begin{array}{lll} g_k^\nu & = \frac{\nu-1}{\nu}\ g_k^{\nu-1} & k\ = 1\ \cdots \nu-1 \\ G_k^\nu & = \frac{\nu-1}{\nu}\ G_k^{\nu-1} & k\ = 1\ \cdots \nu-1 \end{array}$$

- Step 3 Solve

$$\text{Min} \qquad c^T x + \theta^\nu$$

$$\begin{aligned} s.\,t. \quad & Ax = b \\ & \theta^{\nu} + G_k^{\nu}\, x \geq g_k^{\nu} \quad k = 1 \cdots \nu \end{aligned}$$

to obtain $x^{\nu+1}$. Go to Step 1

- The algorithm stops if the change in the objective function is small or if no new dual vectors are added to the set of dual solutions, V.

4. Illustrative Examples

This section presents in detail the beginning of the solution procedure of two illustrative examples in order to show the application of both of the algorithms.

4.1 L-Shaped Method

The example consists on solving the problem:

$$\begin{aligned} \text{Min} \quad & -0.75x + E_u[Q(x,u)] \\ s.\,t. \quad & x \leq 5 \\ & x \geq 0 \end{aligned} \tag{17}$$

where

$$\begin{aligned} Q(x,u) = \text{Min} \quad & -v_1 + 3v_2 + v_3 + v_4 \\ s.\,t. \quad & -v_1 + v_2 - v_3 + v_4 = u + \tfrac{1}{2}\, x \\ & -v_1 + v_2 + v_3 - v_4 = 1 + u + \tfrac{1}{4}\, x \\ & v_1, v_2, v_3, v_4 \geq 0 \end{aligned} \tag{18}$$

and u is defined by a uniform discrete distribution, with $K = 11$ realizations between 0 and -1 {0,-0.1,-0.2,-0.3,-0.4,-0.5,-0.6,-.07,-0.8,-0.9,-1.0}. Each realization has a probability $p_k = 1/11$. The dual of the second stage problem, Equation (18), is given by:

$$\begin{aligned} Q(x,u) = \text{Max} \quad & \pi_1(u + \tfrac{1}{2}x) + \pi_2(1 + u + \tfrac{1}{4}x) \\ s.\,t. \quad & -\pi_1 - \pi_2 \leq -1 \\ & \pi_1 + \pi_2 \leq 3 \\ & -\pi_1 + \pi_2 \leq 1 \\ & \pi_1 - \pi_2 \leq 1 \end{aligned} \tag{19}$$

Solution

- Step 0 $r = s = \nu = 0$
- Step 1 $\nu = 1$, $\theta^1 = -\infty$. Solve:

$$\begin{aligned} \text{Min} \quad & -0.75x \\ s.\,t. \quad & x \leq 5 \\ & x \geq 0 \end{aligned}$$

the solution is $x^1 = 5$.

- Step 2 For $k = 1 \cdots K$ solve:

$$\begin{aligned} z = \text{Min} \quad & v_{k1}^+ + v_{k2}^+ + v_{k1}^- + v_{k2}^- \\ s.\,t. \quad & -v_{k1} + v_{k2} - v_{k3} + v_{k4} + v_{k1}^+ \\ & -v_{k1}^- = u^k + \tfrac{1}{2}\, x^1 \\ & -v_{k1} + v_{k2} + v_{k3} - v_{k4} + v_{k2}^+ \\ & -v_{k2}^- = 1 + u^k + \tfrac{1}{4}\, x^1 \\ & v_{k1}, v_{k2}, v_{k3}, v_{k4} \geq 0 \\ & v_{k1}^+, v_{k2}^+, v_{k1}^-, v_{k2}^- \geq 0 \end{aligned} \tag{20}$$

Recall that $x^1 = 5$. For instance, for $k = 1$, $u^k = 0$ (first discrete value), then problem (20) would become:

$$\begin{aligned} z = \text{Min} \quad & v_{11}^+ + v_{12}^+ + v_{11}^- + v_{12}^- \\ s.\,t. \quad & -v_{11} + v_{12} - v_{13} + v_{14} + v_{11}^+ \\ & -v_{11}^- = 2.5 \\ & -v_{11} + v_{12} + v_{13} - v_{14} + v_{12}^+ \\ & -v_{12}^- = 2.25 \\ & v_{11}, v_{12}, v_{13}, v_{14} \geq 0 \\ & v_{11}^+, v_{12}^+, v_{11}^-, v_{12}^- \geq 0 \end{aligned} \tag{21}$$

The solution to (21) is $z = 0$. The results of all of the problems ($k = 1 \cdots 11$) are summarized in Table 1. Variables not shown are equal to zero. Observe that no feasibility cuts are needed in this iteration ($z = 0$ in all of the cases). As a matter of fact, the example used here is a problem with complete recourse.

- Step 3 For $k = 1 \cdots K$ solve:

$$\begin{aligned} z = \text{Min} \quad & -v_{k1} + 3v_{k2} + v_{k3} + v_{k4} \\ s.\,t. \quad & -v_{k1} + v_{k2} - v_{k3} + v_{k4} = \\ & u^k + \tfrac{1}{2}\, x^1 \\ & -v_{k1} + v_{k2} + v_{k3} - v_{k4} = \\ & 1 + u^k + \tfrac{1}{4}\, x^1 \\ & v_{k1}, v_{k2}, v_{k3}, v_{k4} \geq 0 \end{aligned} \tag{22}$$

Recall that $x^1 = 5$. For instance, for $k = 1$, $u^k = 0$, then Equation (22) would become:

$$z = \text{Min} \quad -v_{11} + 3v_{12} + v_{13} + v_{14}$$

Table 1: Determining feasibility of the second stage.

k	u^k	z^k	v_{k2}	v_{k4}
1	0	0	2.375	0.125
2	-0.1	0	2.275	0.125
3	-0.2	0	2.175	0.125
4	-0.3	0	2.075	0.125
5	-0.4	0	1.975	0.125
6	-0.5	0	1.875	0.125
7	-0.6	0	1.775	0.125
8	-0.7	0	1.675	0.125
9	-0.8	0	1.575	0.125
10	-0.9	0	1.475	0.125
11	-1.0	0	1.375	0.125

Table 2: Obtaining an optimality cut.

k	u^k	z^k	v_{k2}	v_{k4}	π^1_{k1}	π^1_{k2}
1	0	3.5	2.375	0.125	2	1
2	-0.1	3.2	2.275	0.125	2	1
3	-0.2	2.9	2.175	0.125	2	1
4	-0.3	2.6	2.075	0.125	2	1
5	-0.4	2.3	1.975	0.125	2	1
6	-0.5	2.0	1.875	0.125	2	1
7	-0.6	1.7	1.775	0.125	2	1
8	-0.7	1.4	1.675	0.125	2	1
9	-0.8	1.1	1.575	0.125	2	1
10	-0.9	0.8	1.475	0.125	2	1
11	-1.0	0.5	1.375	0.125	2	1

$$\begin{aligned} s.\ t.\ -v_{11} + v_{12} - v_{13} + v_{14} &= 2.5 \\ -v_{11} + v_{12} + v_{13} - v_{14} &= 2.25 \\ v_{11}, v_{12}, v_{13}, v_{14} &\geq 0 \end{aligned} \tag{23}$$

The solution to Equation (23) is $z = 3.5$, $v_{12} = 2.375$, $v_{14} = 0.125$, $\pi^1_{11} = 2$, $\pi^1_{12} = 1$. The results of all of the problems ($k = 1 \cdots 11$) are summarized in Table 2.

Recall that $p_k = 1/11$. Also note that, for instance,

$$\begin{aligned} (\pi^1_1)^T\ h_1 &= \pi^1_{11}\ (u^1) + \pi^1_{12}\ (1 + u^1) \\ (\pi^1_1)^T\ T_1 &= \pi^1_{11}\ (-\tfrac{1}{2}) + \pi^1_{12}\ (-\tfrac{1}{4}) \end{aligned}$$

so that $g_1 = -0.5$ and $G_1 = 1.25$. Therefore, the optimality cut is $\theta \geq -0.5 + 1.2x$. Now we return to Step 1 and begin another iteration.

- Step 1 $\nu = 2$. Solve:

$$\begin{aligned} \text{Min} \quad & -0.75x + \theta \\ s.\ t. \quad & x \leq 5 \\ & \theta \geq -0.5 + 1.25\ x \\ & x \geq 0 \end{aligned} \tag{24}$$

The solution is $x^2 = 0$ and $\theta^2 =$ -0.5. The procedure continues until optimal solution. The solution to the problem is $x = 0.533$ and $\theta = 1.142$.

4.2 Stochastic Decomposition

The example consists of solving again the problem defined by Equations (17) and (18). However, u is now defined by a uniform continuous distribution $(-1, 0)$, so that sampling is necessary. Here, a single-dimension uniform sampling is used.

Solution. Iteration 1

- Step 0 $\nu = 0$, $\theta^0 = -\infty$, $V_0 = \{\emptyset\}$, $x^1 = 0$
- Step 1 $\nu = 1$. From the sampling, $u^1 = -0.3$.
- Step 2

  a) Solve the problem

$$\begin{aligned} \text{Max } & \pi^1_{11}(u^1 + \tfrac{1}{2}x^1) + \pi^1_{12}(1 + u^1 + \tfrac{1}{4}x^1) \\ &= -0.3\ \pi^1_{11} + 0.7\ \pi^1_{12} \\ s.\ t. \quad & -\pi^1_{11} - \pi^1_{12} \leq -1 \\ & \pi^1_{11} + \pi^1_{12} \leq 3 \\ & -\pi^1_{11} + \pi^1_{12} \leq 1 \\ & \pi^1_{11} - \pi^1_{12} \leq 1 \end{aligned}$$

  The solution is $\pi^1_1 = [\pi^1_{11}\ \pi^1_{12}]^T = [1\ 2]^T$. Set $V_1 = V_0 \cup \pi^1_1 = \{(1, 2)\}$

  b) Coefficients of the cut for the first iteration

$$\begin{aligned} g^1_1 = \tfrac{1}{1}[(\pi^1_1)^T h_1] &= \pi^1_{11}(u^1) + \pi^1_{12}(1 + u^1) \\ &= (1)(-0.3) + (2)(0.7) = 1.1 \\ G^1_1 = \tfrac{1}{1}[(\pi^1_1)^T T_1] &= \pi^1_{11}(-\tfrac{1}{2}) + \pi^1_{12}(-\tfrac{1}{4}) \\ &= (1)(-\tfrac{1}{2}) + (2)(-\tfrac{1}{4}) = -1 \end{aligned}$$

  The resulting cut is then $g^1_1 - G^1_1 x = 1.1 + x$.

  c) No updating of cuts is necessary in the first iteration.

- Step 3 Solve:

$$\begin{array}{rc} \text{Min} & -0.75x \ + \ \theta^1 \\ s.\ t. & x \leq 5 \\ & \theta^1 \geq 1.1 \ + \ x \\ & x \geq 0 \end{array}$$

The solution is $x^2 = 0$ and $\theta^1 = 1.1$.

Iteration 2

- Step 1 $\nu = 2$. From sampling $u^2 = -0.9$. Recall $x^2 = 0$ and $u^1 = -0.3$.
- Step 2

a) Solve the problem:

$$\begin{array}{c} \text{Max } \pi_{21}^2(u^2 + \frac{1}{2}x^2) + \pi_{22}^2(1 + u^2 + \frac{1}{4}x^2) \\ = \ -0.9\ \pi_{21}^2 \ + \ 0.1\ \pi_{22}^2 \\ s.\ t. \quad -\pi_{21}^2 \ - \pi_{22}^2 \leq \ -1 \\ \pi_{21}^2 \ + \ \pi_{22}^2 \leq 3 \\ -\pi_{21}^2 \ + \pi_{22}^2 \leq 1 \\ \pi_{21}^2 \ - \pi_{22}^2 \leq 1 \end{array}$$

The solution is $\pi_2^2 = [\pi_{21}^2 \ \ \pi_{22}^2]^T = [0 \ \ 1]^T$. Set $V_2 = V_1 \cup \pi_2^2 = \{(1,2),(0,1)\}$

b) Get the coefficients of the cut for the second iteration. It is necessary to calculate π_1^2 by using the first sample u^1. Solve:

$$\begin{array}{c} \text{Max } \pi_{11}^2(u^1 + \frac{1}{2}x^2) + \pi_{12}^2(1 + u^1 + \frac{1}{4}x^2) \\ = \ -0.7\ \pi_{11}^2 \ + \ 0.3\ \pi_{12}^2 \\ s.\ t. \quad \pi_1^2 \ \in \ V_2 = \pi_1^2 \in \{(1,2),(0,1)\} \end{array}$$

The solution is $\pi_1^2 = [\pi_{11}^2 \ \ \pi_{12}^2]^T = [1 \ \ 2]^T$. Then, the coefficients of the optimality cut for the second iteration are $g_2^2 = \frac{1}{2}\{[1(-0.3)+2(0.7)]+[0(-0.9)+1(0.1)]\} = 0.6$ and $G_2^2 = \frac{1}{2}\{[1(-\frac{1}{2})+2(-\frac{1}{4})]+[0(-\frac{1}{2})+1(-\frac{1}{4})]\} = -0.625$
The resulting cut is then $g_2^2 - G_2^2 x = 0.6 + 0.625x$.

c) Update previous cuts

$$\begin{array}{c} g_1^2 = \frac{1}{2}g_1^1 = 0.55, \ G_1^2 = \frac{1}{2}G_1^1 = -0.5 \\ g_1^2 - G_1^2 x = 0.55 + 0.5x \end{array}$$

- Step 3 Solve the problem:

$$\begin{array}{rc} \text{Min} & -0.75x \ + \ \theta^2 \\ s.\ t. & x \leq 5 \\ & \theta^2 \geq 0.55 \ + \ 0.5x \\ & \theta^2 \geq 0.6 \ + \ 0.625x \\ & x \geq 0 \end{array}$$

the solution is $x^3 = 5$ and $\theta^2 = 3.725$. The procedure continues in the same way. The solution to the problem after 1000 samples is $x = 0.4646$ and $\theta = 1.077744$.

5. Conclusion

This article presented the fundamentals and a description of the two main algorithms for solving $SLPwR$ problems: The L-Shaped method and the Stochastic Decomposition method. Two examples were used to illustrate in detail the application of each of the algorithms.

REFERENCES

[1] Birge, J. R. and F. Louveaux, *A multicut algorithm for two-stage stochastic linear programming* , European Journal of Operations Research , 1988, 34, 384-392.

[2] Birge, J. R. and F. Louveaux, *Introduction to stochastic programming*, Springer-Verlag, New York, 1997.

[3] Bouza, C., *Stochastic programming: the state of the art*, Revista Investigacion Operacional, 1993, 14(2).

[4] Higle, J. L. and S. Sen, *Stochastic decomposition: an algorithm for two-stage linear programs with recourse*, Mathematics of Operations Research, 1991, 16(3): 650-669.

[5] Higle, J. L. and S. Sen, *Stochastic decomposition*, Kluwer Academic Publisher, 1996.

[6] Haneveld, W. K. K. and M. H. vander Vlerk, *Stochastic integer programming: general models and algorithms*, Annals of Operational Research, 1999, 85, 39-57.

Combinatorial Optimization under Uncertainty

Ki-Joo Kim

CUSTOM, Carnegie Mellon University, Pittsburgh, PA 15213 (kijoo@andrew.cmu.edu)[1]

1. Introduction

Stochastic integer programming (SIP) refers to that branch of integer programming problems where there are uncertainties involved in the data or model.

The main difficulty of stochastic programming stems from evaluating the uncertain functions and their expectations. A common method to propagate the uncertainties is to use a *sampling method*, and a generalized stochastic framework for solving optimization under uncertainty problems involves two recursive loops: (1) the inner sampling loop and (2) the outer optimization loop (Figure 1). Since propagating uncertainties and evaluating the uncertain functions are computationally very intensive, this article presents a computationally efficient SIP algorithm based on a new sampling method, called Hammersley sequence sampling (HSS), in the inner sampling loop and the interaction between the inner sampling loop and the outer optimization loop.

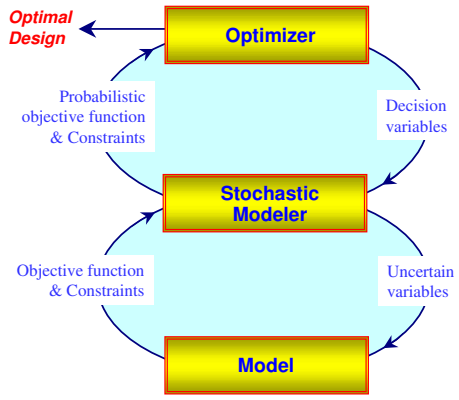


Figure 1: Pictorial representation of stochastic programming framework.

2. Stochastic Annealing - Theory

The stochastic annealing (STA) algorithm [1, 2, 7] is a variant of the simulated annealing [6], and is an algorithm designed to efficiently optimize stochastic integer programming problems. In the stochastic annealing algorithm, the optimizer not only obtains the decision variables but also the number of samples required for the stochastic model. Furthermore, the stochastic annealing algorithm reduces the CPU time by balancing the trade-off between computational efficiency and solution accuracy by the introduction of a penalty function in the objective function. This is necessary, since at high temperatures the algorithm is mainly exploring the solution space and does not require precise estimates of any probabilistic function. The algorithm must select a greater number of samples as the solution nears the optimum.

Annealing temperature schedule (more precisely, cooling schedule) is used to decide the weight ($b(t)$) on the penalty term for imprecision in the probabilistic objective function. The choice of a penalty term also depends on the error bandwidth (ϵ) of the function that is optimized, and must incorporate the effect of the number of samples. The new objective function in stochastic annealing therefore consists of a probabilistic objective value P and the penalty function ($b(t)\epsilon$), which is represented as follows:

$$\min Z(cost) = P(x;u) + b(t)\epsilon. \qquad (1)$$

The weighting function $b(t)$ can be expressed in terms of the temperature levels (t), and is given by $b(t) = b_o/k^t$ where b_o and k are constants. The error bandwidth of the MCS samples (ϵ_{MCS}) is estimated from the central limit theorem while the error bandwidth of the HSS samples (ϵ_{HSS}) is determined from a fractal dimension analysis [2, 3].

3. Efficiency Improvements in the HSTA Algorithm

As SA is a probabilistic method, several random probability functions are involved in this algorithm. The random probability A_{ij} is used for acceptance determination in Metropolis criterion while the random generation probabilities G_{ij} are used to generate subsequent configurational moves. It is known that SA is affected little by the use of different acceptance probability distributions [8]. However, G_{ij} for generating configuration j from i can significantly affect the overall efficiency of the annealing process.

[1]Research supported by the National Science Foundation Goali Project (CTS-9729074).

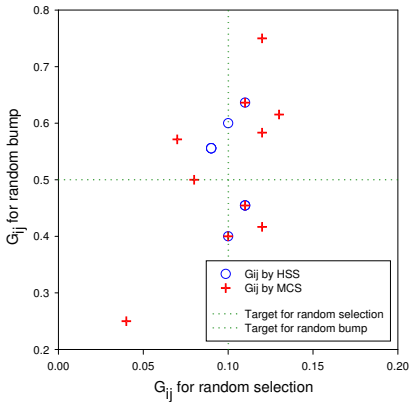


Figure 2: G_{ij} from MCS and HSS.

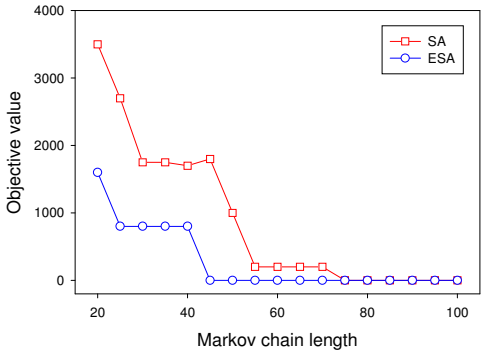


Figure 3: Objective trajectories of ESA and SA.

The G_{ij} of the conventional SA algorithms rely on *pseudo*-random number generators such as MCS, which results in clustered moves over the configuration surface. Therefore, a larger number of moves or generations are required to cover the whole configuration surface evenly, and this results in a larger number of moves (i.e., *Markov chain length*) at each temperature level. The HSS technique, a *quasi*-random number generator, can generate uniform samples over the k-dimensional hypercube. In this work, we have used the HSS technique for the generation probabilities G_{ij} to develop a new SA algorithm called efficient simulated annealing (ESA).

Figure 2 shows G_{ij} probabilities of HSS and MCS for the following test function, $f(y) = \sum_{i=1}^{10} y_i^2$. Because there are 10 elements in the discrete decision vector $\mathbf{y}$, the ideal probability of selecting any element is 0.1, and the value of a selected element can be randomly bumped up or down with a probability of 0.5. The circle and cross symbols in this figure are for the G_{ij} values from HSS and MCS, respectively. Since HSS can generate more uniform samples in the multivariate space, G_{ij} from HSS is closer to the ideal probabilities than G_{ij} from MCS. Thus HSS requires less number of moves to approximate the ideal probabilities.

Figure 3 shows trajectories of the objective value for the test function with different Markov chain lengths. ESA found the global solution with a Markov chain length of 45 at each temperature while the traditional SA exploited a Markov chain length of 75 to reach the same solution. Thus, ESA provides a significant reduction in moves at each temperature, and ESA is approximately 30 $\sim$ 54 % more efficient than the conventional SA [4].

The same idea of using the HSS technique for the generation probability G_{ij} can be applied to STA. The efficient stochastic annealing (ESTA) algorithm integrates the HSS for the generation probability, but still uses the central limit theorem to evaluate the sampling errors. A new variant of stochastic annealing, HSTA (Hammersley stochastic annealing), therefore, incorporates (i) HSS for the generation probability G_{ij}, (ii) HSS in the inner sampling loop for N_{samp} determination, and (iii) the HSS-specific error bandwidth (ϵ_{HSS}) in the penalty term.

Table 1: Efficiency improvements.

Stochastic algorithm	*Total moves*
SA + fixed N_{samp}	274,200
ESA + fixed N_{samp}	170,000
STA	5,670
ESTA	3,265
HSTA	1,793
Test function : $z = \sum_{i=1}^{10} (u_i\, x_i - \frac{i}{10})^2 + \sum_{i=1}^{10} (u_i\, y_i^2) - \prod_{i=1}^{10} \cos(4\pi\, u_i\, y_i)$	

Table 1 shows total number of configurational moves of different stochastic optimization methods. The first two algorithms are (conventional) stochastic optimization algorithms with a fixed N_{samp} while the last three algorithms are stochastic annealing algorithms with a varying N_{samp}. We can see that there is significant improvement when we use the STA algorithms instead of the conventional stochastic optimization methods. In addition, HSTA is 68% more efficient that the basic STA algorithm.

4. Numerical Example

This section describes step-by-step simulations of the HSTA algorithm. The test function is

$$\begin{array}{ll} \min & \sum_{i=1}^{2}\left(u_i \times y_i^2\right) \\ & -20 \le \mathbf{y} \le 20 \\ & \mathbf{u} \sim N(0.5, 0.16) \end{array} \quad (2)$$

The initial configuration of $\mathbf{y}$ is (20,20), and the uncertainty variable $\mathbf{u}$ follows normal distribution with a mean of 0.5 and a standard deviation of 0.16.[2] The simulation conditions for HSTA are summarized in the following Table 2.

Table 2: HSTA conditions.

Initial temperature	1
Temperature decrement (α)	0.85
Markov chain length (I)	20
Initial N_{samp}	30
b_0	0.005
k	0.940

Step 1 is to generate 20 (i.e., Markov chain length) sets of six random numbers using HSS. Two random numbers are used for the G_{ij}, three random numbers (H_k) for determining N_{samp}, and the A_{ij} for the Metropolis criteria.

Step 2 is to generate a next configuration. If $G_{ij,1}$ for random selection is less than 0.5, then y_1 is selected for random bump. If $G_{ij,2}$ for random bump is less than 0.5, then the value of the selected y_i is decreased. If the bumped value resides outside the bounds, the random bump is increased to the original y_i value. In this example, since the G_{ij} are 0.2031 and 0.6914, y_1 is increased to 21. But this is outside the bounds, and thus y_1 becomes 19.

Step 3 is to determine N_{samp} for trade-off between efficiency and solution accuracy. To determine N_{samp}, three random numbers (H_k) generated by the HSS technique are used. If H_1 is less than 0.5, then $N_{\text{samp}} + 5 \times H_2$ becomes a new N_{samp}. Otherwise, $N_{\text{samp}} - 5 \times H_3$ becomes a new one. The new N_{samp} is 28.

Step 4 is to generate N_{samp} samples for the uncertain variable $\mathbf{u}$ using HSS. Then the probabilistic objective, constraints, expected value, and penalty function are evaluated. The expected value of the objective is 219.36, and the penalty term is 1.3E-5. We can see that the penalty term is very small as compared to the expected value because of a HSS-specific ϵ.

Table 3: Configurations at the first TL.

I	y_1	y_2	$G_{ij,1}$	$G_{ij,2}$	N_{samp}	*Penalty*	$E[z]$
0	20	20					800.00
1	19	20	0.2031	0.6914	28	1.3E-5	219.36
2	19	19	0.8281	0.3580	27	1.4E-5	210.86
3	18	19	0.3281	0.0247	31	1.1E-5	196.78
4	18	20	0.5781	0.9753	35	0.9E-5	208.18
5	18	19	0.0781	0.6420	30	1.2E-5	208.06
6	18	18	0.8906	0.3086	26	1.5E-5	189.02
7	19	18	0.3906	0.8642	21	2.2E-5	202.63
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
19	15	16	0.3672	0.7160	27	1.4E-5	140.30
20	14	15	0.6172	0.3827	24	1.7E-5	121.28

Step 5 is to determine if the current configuration is accepted or not based on the Metropolis criterion. Since $\Delta E[z] = E[z]_{\text{new}} - E[z]_{\text{old}}$ is negative, the current configuration (19,20) is accepted.

Step 6 is to repeat the Steps from 2 to 5 if the current iteration point is smaller than Markov chain length. Table 3 shows this full 20 iterations at the first temperature level (TL).

Step 7 is to check the stopping criteria and decrease temperature. If any of the stopping criteria is satisfied, then the simulation is terminated with a successful result. Otherwise, the new temperature becomes $T = \alpha T$, and simulation goes back to Step 2. Table 4 shows the simulation results with respect to the temperature level (TL). The optimum solution is found at the 10th temperature level. This table also shows the STA simulation results for comparison. Since MCS is used for the G_{ij} and the error bandwidth, the STA requires a greater number of temperature levels (i.e., 15 levels in this table).

5. Conclusion

This article presented hierarchical improvements in the SA based algorithms for solving large-scale combinatorial optimization problems under uncertainty, and also presented step-by-step example calculations of the HSTA algorithm. The HSTA algorithm incorporates the uniformity property and fast conver-

[2]The 0.001 and 0.999 quantiles of this distribution are 0 and 1.

Table 4: HSTA and STA simulation results.

	HSTA				STA with MCS			
TL	*E[z]*	$b(t)\epsilon$	y_1	y_2	*E[z]*	$b(t)\epsilon$	y_1	y_2
1	121.28	2E-5	14	15	140.82	0.126	16	15
2	53.41	4E-5	9	10	80.92	0.104	13	10
3	12.04	4E-5	4	5	38.40	0.074	10	5
4	0.00	4E-5	0	0	16.99	0.040	7	2
5	0.59	3E-5	1	-1	5.46	0.015	4	1
6	0.59	3E-5	1	-1	0.32	0.001	1	0
7	0.31	3E-5	1	0	0.32	0.001	-1	0
8	0.31	3E-5	1	0	0.32	0.001	-1	0
9	0.31	4E-5	1	0	0.61	0.001	-1	-1
10	0.00	5E-5	0	0	0.61	0.001	-1	-1
11					0.61	0.002	-1	-1
12					0.61	0.002	-1	-1
13					0.61	0.002	-1	-1
14					0.61	0.002	-1	-1
15					0.00	0.000	0	0

gence property of the HSS technique, and the error bandwidth based on the HSS technique. Thus, it is found that the HSTA algorithm is 68% more efficient than the conventional STA algorithm. The HSTA algorithm can be a useful tool for large-scale combinatorial stochastic programming problems, and a real world case study of computer-aided molecular design under uncertainty can be found in the author's paper [5].

REFERENCES

[1] Chaudhuri, P. and U.M. Diwekar, *Process synthesis under uncertainty: A penalty function approach*, AIChE Journal, 1996, 42(3): 742-752.

[2] Chaudhuri, P. and U.M. Diwekar, *Synthesis approach to the determination of optimal waste blends under uncertainty*, AIChE Journal, 1999, 45(8): 1671-1687.

[3] Diwekar, U.M., *An efficient approach to optimization under uncertainty*, Computational Optimization and Applications (submitted), 2001.

[4] Kim, K.-J. and U.M. Diwekar, *Efficient combinatorial optimization under uncertainty – Part I. Algorithmic development (accepted)*, Ind. & Eng. Chem. Res., 2001.

[5] Kim, K.-J. and U.M. Diwekar, *Efficient combinatorial optimization under uncertainty – Part II. Application to stochastic solvent selection (accepted)*, Ind. & Eng. Chem. Res., 2001.

[6] Kirkpatrick, S., C.D. Gellat, Jr., and M.P. Vecchi, *Optimization by simulated annealing*, Science, 1983, 220: 671-680.

[7] Painton, L.A. and U.M. Diwekar, *Stochastic annealing for synthesis under uncertainty*, European Journal of Operational Research, 83: 489-502.

[8] van Laarhoven, P.J.M. and E.H.L. Aarts, Simulated Annealing: Theory and Applications, Reidel Publishing Company, 1987.

Stochastic Nonlinear Optimization - The BONUS Algorithm

Kemal H. Sahin

CUSTOM, Carnegie Mellon University, Pittsburgh, PA 15213 (khsahin@andrew.cmu.edu).[1]

1. BONUS Background

This article describes a new algorithm for the optimization of nonlinear, uncertain systems developed by our research center [1]. General techniques for these types of optimization problems determine a statistical representation of the objective, such as maximum expected value or minimum variance. Several decomposition algorithms have been developed for linear problems; however, nonlinear formulations are solved through evaluating the model for a series of samples. Once embedded in an optimization framework, the iterative loop structure emerges where decision variables are determined, a sample set based on these decision variables is generated, the model is evaluated for each of these sample points, and the probabilistic objective function value and constraints are evaluated, as shown by the black arrows in the center of Figure 1. When one considers that nonlinear optimization techniques rely on an objective function and constraints evaluation for each iteration, along with derivative estimation through perturbation analysis, the sheer number of model evaluations rises significantly to render this approach ineffective for even moderately complex models.

The Better Optimization of Nonlinear Uncertain Systems (BONUS) algorithm, indicated in Figure 1 by the thick grey arrows, samples the solution space of the objective function at the beginning of the analysis by using a base distribution covering the entire feasible range. As decision variables change, the underlying distributions for uncertain values change, and the proposed algorithm estimates the objective

[1]Research supported by Sandia National Laboratories.

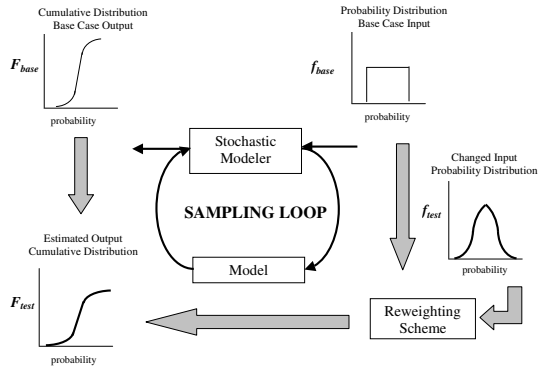


Figure 1: Density estimation approach to optimization under uncertainty.

function value based on the ratios of the probabilities for the current and the base distributions, which are approximated using kernel density estimation (KDE) techniques. Thus, BONUS avoids sample model runs in subsequent iterations.

The algorithm can be summarized in the following outline:

1. Generate base samples and calculate respective KDEs, evaluate model for each base sample.
2. Optimization.
    a. Generate sample around the decision variables.
    b. Determine KDEs for each sample point with respect to base distribution.
    c. Estimate objective function value through reweighting scheme.
    d. SQP or similar state of the art nonlinear optimization approaches that rely on quasi-Newton methods perturb each decision variable to estimate gradient information. Use same approach as in steps (a)-(c) to estimate the objective function value using KDE and reweighting at the perturbed point.
    e. Optimizer generates new vector of decision variables, repeat until convergence.

As seen above, the model is only evaluated for the base sample, the objective function value is then estimated using the base sample data and the respective probabilities for the base sample points and the samples generated during optimization.

2. Illustrative Numerical Example

To illustrate the concepts behind the BONUS algorithm, we present a numerical example to explain the approach:

2.1 Model

$$\min E[Z] = E[(\tilde{x}_1 - 7)^2 + (\tilde{x}_2 - 4)^2] \quad (1)$$

$$\text{s.t.} \quad \tilde{x}_1 \in N[\mu = x_1^\star, \sigma = 0.033 \cdot x_1^\star] \quad (2)$$

$$\tilde{x}_2 \in U[0.9 \cdot x_2^\star, 1.2 \cdot x_2^\star] \quad (3)$$

$$4 \leq x_1 \leq 10 \quad (4)$$

$$0 \leq x_2 \leq 5 \quad (5)$$

Here, E represents the expected value, and the goal is to minimize the mean of the objective function calculated for two uncertain decision variables, x_1 and x_2. The optimizer determines the value $x_1^\star$, which has an underlying normal distribution with $\pm$ 10% of the nominal value of $x_1^\star$ as the upper and lower 0.1% quantiles. Similarly, $\tilde{x}_2$ is uniformly distributed around $x_2^\star$, with cutoff ranges at $[-10\%, +20\%]$.

Step 1: The first step in BONUS is determining the base distributions for the decision variables, followed by generating the output values for this model. In order to capture the entire possible range of uncertain variables, these base distributions have to cover the entire range, including variations. For instance, for x_2, the range extends to $(0 \cdot 0.9) \leq x_2 \leq (5 \cdot 1.2)$ to account for the uniformly distributed uncertainty. Due to space limitations, the illustrative presentation of the kernel density and reweighting approach is performed for a sample size of 10, while the remainder of the work uses $N = 100$ samples. A sample realization using Monte Carlo sampling is given in Table 1.

After this sample is generated, KDE for the base sample is applied to determine the probability of each sample point with respect to the sample set. This is performed for each decision variable separately by approximating each point through a Gaussian kernel, and adding these kernels to generate the probability distribution for each point, as given in

Table 1: Base sample.

Sample No.	x_1	x_2	Z
1	5.6091	0.3573	15.2035
2	3.7217	1.9974	14.7576
3	6.2927	4.2713	0.5738
4	7.2671	3.3062	0.5527
5	4.1182	1.3274	15.4478
6	7.7831	1.5233	6.7472
7	6.9578	1.1575	8.0818
8	5.4475	3.6813	2.5119
9	8.8302	2.9210	4.5137
10	6.9428	3.7507	0.0654
Mean	6.2970	2.4293	—
Std. Dev	1.5984	1.3271	—

Equation 6 [2].

$$\hat{f}(x_i(k)) = \frac{1}{N \cdot h} \sum_{j=1}^{N} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x_i(k)-x_i(j)}{h}\right)^2} \quad (6)$$

Here, h is the width for the Gaussian kernel and depends on the variance σ and sample size N of the data set as follows:

$$h = 1.06 \cdot \sigma \cdot N^{-\frac{1}{5}} \quad (7)$$

For our example, $h(x_1) = 1.06 \cdot 1.5984 \cdot 10^{-0.2} = 1.0690$ and $h(x_2) = 1.06 \cdot 1.3271 \cdot 10^{-0.2} = 0.8876$. Using the first value, one can calculate $\hat{f}(x_1(1)) = \frac{1}{10 \cdot 1.0690} \sum_{j=1}^{10} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{5.6091-x_1(j)}{1.0690}\right)^2} = 0.1769$. This step is repeated for every point, resulting in the kernel density estimates provided in Table 2.

Table 2: Base sample kernel density estimates.

x_1	$\hat{f}(x_1)$	x_2	$\hat{f}(x_2)$
5.6091	0.1769	0.3573	0.1277
3.7217	0.0932	1.9974	0.2114
6.2927	0.2046	4.2713	0.1602
7.2671	0.2000	3.3062	0.2190
4.1182	0.1110	1.3274	0.2068
7.7831	0.1711	1.5233	0.2117
6.9578	0.2090	1.1575	0.1992
5.4475	0.1691	3.6813	0.2100
8.8302	0.0920	2.9210	0.2152
6.9428	0.2092	3.7507	0.2063

Step 2: All these steps were preparations for the optimization algorithm, where repeated calculations of the objective function will be bypassed through the reweighting scheme.

Step 2a: For the first iteration, assume that the initial value for the decision variables is $x_1 = 5$ and $x_2 = 5$. For these values, another sample set is generated, as shown in Table 3, accounting for the uncertainties described in Equations 2 and 3.

Table 3: Sample - optimization iteration 1.

Sample No.	$\tilde{x}1$	$\tilde{x}_2$
1	4.7790	5.7625
2	4.9029	5.5740
3	5.0347	5.9199
4	4.9686	5.8697
5	4.9001	5.9967
6	4.9819	5.1281
7	5.0316	5.4877
8	5.0403	5.4841
9	4.9447	5.7557
10	5.0344	4.7531
Mean	4.9618	5.5731
Std. Dev	0.0836	0.3862

The expected value of Z, which is to be minimized during optimization, is calculated as 6.7695. This value will be estimated using a reweighting approach [3], given in Steps 2b and 2c.

Step 2b: Now, the KDE for the sample $(f(x_i))$ generated around the decision variables has to be calculated. The Gaussian kernel width $h(\tilde{x_1}) = 1.06 \cdot 0.0837 \cdot 10^{-0.2} = 5.598 \cdot 10^{-2}$. Using this value, one can calculate $f(x_1(1)) = \frac{1}{10 \cdot 5.598 \cdot 10^{-2}} \sum_{j=1}^{10} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{5.609-\tilde{x}_1(j)}{5.598 \cdot 10^{-2}}\right)^2} = 5.125 \cdot 10^{-23}$. Again, this step is repeated for every point of the sample with respect to the base distribution data, resulting in the kernel density estimates provided in Table 4.

Step 2c: Using these and the base KDE values, weights are calculated for each sample point j as

$$\omega_j = \frac{f(x_1(j))}{\hat{f}(x_1(j))} \cdot \frac{f(x_2(j))}{\hat{f}(x_2(j))}, \quad j = 1, ..., N \quad (8)$$

In our illustrative example, the only two non-zero weights are $\omega_5 = 1.699 \cdot 10^{-68}$) and $\omega_8 = 4.152 \cdot 10^{-15}$. These weights are normalized and multiplied with the output of the base distribution to estimate the

Table 4: Optimization iteration 1 - KDE.

No	x_1	$f(x_1)$	x_2	$f(x_2)$
1	5.6091	$5.125{\cdot}10^{-23}$	0.3573	0
2	3.7217	0	1.9974	$2.989{\cdot}10^{-26}$
3	6.2927	0	4.2713	$2.777{\cdot}10^{-2}$
4	7.2671	0	3.3062	$2.376{\cdot}10^{-8}$
5	4.1182	$3.918{\cdot}10^{-31}$	1.3274	$9.958{\cdot}10^{-40}$
6	7.7831	0	1.5233	$1.745{\cdot}10^{-35}$
7	6.9578	0	1.1575	$1.303{\cdot}10^{-43}$
8	5.4475	$5.218{\cdot}10^{-12}$	3.6813	$2.826{\cdot}10^{-5}$
9	8.8302	0	2.9210	$1.844{\cdot}10^{-12}$
10	6.9428	0	3.7507	$8.311{\cdot}10^{-5}$

objective function value:

$$E^{est}[Z] = \sum_{j}^{N} \overline{\omega_j} \cdot Z(j) \tag{9}$$

For our illustrative example, this reduces to

$$E^{est}[Z] = \overline{\omega_8} \cdot Z(8) = 1.0000 \cdot 2.5119 = 2.5119 \tag{10}$$

as the normalization eliminates all but one weight. Note that this illustrative example was developed with an unrealistically small sample size. Hence, the efficiency of the estimation technique cannot be judged from this example. Further, due to the inaccuracy of the estimate resulting from the small sample size, we will not present results for Steps 2d and 2e for just 10 samples, but use 100 samples. Also note that Steps 2d and 2e basically repeat the procedures in Steps 2a through 2c for a new sample set around a perturbed point, for instance $x_1 + \Delta x_1 = 5 + 0.001 \cdot 5 = 5.005$.

Table 5: Optimization progress at $N = 100$.

Iteration	x_1	x_2	$E^{est}[Z]$
0	5.000	5.000	5.958
1	9.610	2.353	9.238
2	7.065	3.814	0.258

The results obtained using the BONUS algorithm for optimization converge to the same optimal solution as obtained using a 'brute force' analysis normally used in stochastic NLPs where the objective is calculated for each iteration by calculating the objective function value for each generated sample point. In this example BONUS used only 100 model runs, while the brute force optimization evaluated the model 600 times for the two iterations.

The BONUS Algorithm has been applied to extend studies in off-line quality control of continuously stirred tank reactors to include optimization, and to capacity expansion studies for electric utilities. For further information, please contact the author at `khsahin@andrew.cmu.edu`.

REFERENCES

[1] Sahin, K.H. and U.M. Diwekar, *A New Algorithm for Stochastic Nonlinear Programming: A Reweighting Approach for Derivative Approximation*, Session 290, Annual Meeting, American Institute of Chemical Engineers (AIChE), Reno, NV, 2001.

[2] Silverman, B.W, Density Estimation for Statistics and Data Analysis, Chapman & Hall (CRC Reprint), Boca Raton, USA, 1998.

[3] Hesterberg, T., *Weighted Average Importance Sampling Defensive Mixture Distributions*, Technometrics, 1995, 37: 185.

[4] Kalagnanam, J.R. and U.M. Diwekar, *An Efficient Sampling Technique for Off-line Quality Control*, Technometrics, 1997, 39: 308.

Bulletin

Call for Papers for Special Issue of Annals of Operations Research on "Applied Optimization under Uncertainty"

Robust decision making under uncertainty is of fundamental importance in numerous disciplines and application areas. For many practical issues, decision making involves multiple, often conflicting, goals and poses a challenging and complex optimization problem. Recent innovations in the techniques underlying multi-objective optimization, in the characterization of uncertainties, and in the ability to develop and apply these methods outside of traditional application domains greatly enhances their utility and promise. The main focus of this special issue of Annals of Operations Research on "Applied Optimization Under Uncertainty" is to provide research

papers related to systematic algorithms, methods, and approaches for rapid and reliable multi-objective decision making under uncertainty and to successfully apply these methods in diverse application areas. The wide scope of this area can be overwhelming and mandates extensive interactions between various disciplines both in the development of the methods and in their fruitful application to real world problems. The issue is dedicated to bringing together papers on the development of the theory of optimization under uncertainty methods and applications to practical problems. Papers on perspectives and overviews that pose new challenges for the development of the theory and methods are also encouraged.

Annals of Operations Research is one of the most renowned journals in the field. The Center for Uncertain Systems: Tools for Optimization and Management (CUSTOM), Carnegie Mellon University is holding a mini-conference, sponsored by Sandia National Laboratories, on the topic of "Applied Optimization and Under Uncertainty," in December 2001. This special issue and mini-conference provide an excellent opportunity to advance the knowledge of this evolving and widely applicable area.

AREAS OF INTEREST. For the special issue, we expect original papers of high quality providing theoretical and/or computational results for algorithms and applications of optimization under uncertainty, and multi-objective optimization. We invite papers in the following general areas:

1. Algorithms, Methods, and Tools for Multi-objective, Stochastic Programming and Optimization
2. Challenges and Opportunities for Multi-objective, Stochastic Programming and Optimization
3. Materials and Molecular Modeling
4. Manufacturing, Planning, and Management
5. Energy and Environment

REVIEWING. The submitted papers will be peer-reviewed in the same manner as any other submission to a leading international journal. The major acceptance criterion for a submission is the quality and originality of the contribution.

SUBMISSION. The deadline for submission is May 1, 2002. (This is a Sunday.)

Manuscripts must be written in English and should be submitted electronically in a platform-independent format such as postscript or pdf. Please send your submission to custom@cmu.edu. Please follow the instructions for authors for Annals of Operations Research.

Best Regards,

Urmila Diwekar, CUSTOM, Carnegie Mellon University, Pittsburgh, PA 15213.
Reha Tutuncu, CUSTOM, Carnegie Mellon University, Pittsburgh, PA 15213.
Andrew Schaefer, CUSTOM, University of Pittsburgh, PA 15261.

Comments from the Chair and Editor

Seventh SIAM Conference on Optimization

Greetings to all members of SIAG/OPT and readers of the Newsletter.

After Sept 11, many of us contacted each other by email just to say hello to friends. As one email message stated: "It is hard to imagine the hatred that could lead people to do this." and: "Give hugs to everyone you know today!" and: "The world has changed forever today." The events following Sept. 11 were also harrowing and stressful. The organizers (including me) of the upcoming conference on optimization (in May in Toronto) were expecting a poor turnout, with few people risking airtravel. However, I am happy to announce that this is *not* the case. The program schedule for the Seventh Siam Conference on Optimization is ready and it is *full*; talks start at 8:15AM and end at 7PM (followed by receptions, poster sessions and meetings for some of us). One problem was that there were not enough slots for all the contributed talks. (Thanks go to Ariela Sofer and Tom Coleman for all the hard work involved in making up the schedule.) We (the organizers) hope to see you all in Toronto this spring.

Two short courses (on *Numerical Optimization - Algorithms and Software* and on *Automatic*

Differentiation) immediately preceed the conference, on Sunday May 19. Sunday is also *Don Goldfarb Day*, with Technical Sessions from 5:15 PM to 7:00 PM and a Banquet at 7:30 PM. In addition, there will be a group of visitors (in optimization) before/during/after this SIAM Conference on Optimization. The group will be visiting the Fields Institute Toronto, Ontario, see URL: www.fields.utoronto.ca/. This is part of the Thematic Year on Numerical and Computational Challenges in Science and Engineering (NCCSE) from August 2001 to July 2002. A schedule of visitors and talks is available with URL:
http://orion.math.uwaterloo.ca/ hwolkowi/henry/ reports/talks.d/t02talks.d/02optfields.d/group.html
Following the conference, there will also be two (Fields) workshops:
May 23 - 25, 2002, Workshop on Validated Computing, Harbour Castle Hotel, Toronto, Organizers: George Corliss, Ken Jackson, Baker Kearfott, Vladik Kreinovich, Weldon Lodwick;
May 27 - 31, 2002, Informal Working Group on Validated Methods for Optimization , Organizers: George Corliss, Tibor Csendes, Ken Jackson, Baker Kearfott.

SIAM 50th Anniversary and 2002 Annual Meeting The annual meeting this summer is also a celebration of SIAM's 50th birthday. This special annual meeting will look at the strides made by industrial and applied mathematics during the past 50 years. The meeting themes cover SIAM's interests, including optimization.

Miscellaneous We need to thank Juan Meza again for his excellent job as editor of our Newsletter. This is the last issue edited by him. I am pleased to announce that Jos Sturm has accepted the SIAG/OPT committee's invitation to be the new editor.

The SIAG/OPT Web site, URL: www.siam.org/siags/siagopt.htm,
points to our own SIAG/OPT webpage, handled by Natalia Alexandrov URL: mdob.larc.nasa.gov/staff/natalia/siagopt/.
We will continue to make this web page useful and interesting.

We continue to encourage members with home pages to register their home pages with SIAM. If you wish to be listed, send a message to Laura Helfrich at helfrich@siam.org with your name and the URL for your Web page.

The e-mail forum continues. You can use this forum for technical questions, announcements of papers, conferences, books, and software. In particular, technical questions are encouraged. Perhaps, this will lead to interesting discussions. You can use this forum by sending a message to opt@siam.org.

Here is one question that I find interesting. The class of problems that we can solve today in optimization has grown tremendously, due both to software and hardware. In particular, the size of problems has grown and the speed has decreased dramatically. The motivation behind many papers now is *larger and faster*. However, there seems to be much less emphasis put on the quality of the solution. (Though some work on robust optimization is being done.)

> Question: Which algorithms can solve large classes of large scale problems robustly? Which property is more important for an algorithm: speed or robustness?

Multi-Media

1. Optimization Online deserves being mentioned again, URL: www.optimization-online.org/. This is a repository for eprints.

2. Eric Weisstein's World of Mathematics is (back) at: mathworld.wolfram.com/. This is definitely worth a look - every day.

3. The Mathematics Genealogy Project is at: hcoonce.math.mankato.msus.edu/. The intent of this project is to compile information about *all* the mathematicians of the world. They solicit information from all schools who participate in the development of research level mathematics and from all individuals who may know desired information. (Go and research your genealogy tree!)