Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 28 (2004) 767–773
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Improving convergence of the stochastic decomposition algorithm by using an efficient sampling technique

Jose M. Ponce-Ortega [a], Vicente Rico-Ramirez [a,*], Salvador Hernandez-Castro [b], Urmila M. Diwekar [c]

[a] *Dept. de Ingenieria Quimica, Instituto Tecnologico de Celaya, Ave. Tecnologico y Garcia Cubas S/N, C.P. 38010 Celaya, Gto., Mexico*
[b] *Universidad de Guanajuato, Col. Noria Alta S/N, C.P. 36050 Guanajuato, Gto., Mexico*
[c] *Department of Chemical Engineering, University of Illinois at Chicago, 810 South Clinton Street, 209 CHB, M/C 110, Chicago, IL 60607, USA*

Abstract

This work focuses on the basic stochastic decomposition (SD) algorithm of Higle and Sen [J.L. Higle, S. Sen, Stochastic Decomposition, Kluwer Academic Publishers, 1996] for two-stage stochastic linear programming problems with complete recourse. The algorithm uses sampling when the random variables are represented by continuous distribution functions. Traditionally, this method has been applied by using Monte Carlo (MC) sampling to generate the samples of the stochastic variables. However, Monte Carlo methods can result in large error bounds and variance. Hence, some other approaches use importance sampling to reduce variance and achieving convergence faster that the method based on the Monte Carlo sampling technique. This work proposes an improvement on this respect. Hence, we propose to replace the use of the Monte Carlo sampling technique or the importance sampling in the SD algorithm by the use of the novel Hammersley sequence sampling (HSS) technique. Recently, such a technique has proved to provide better uniformity properties than other sampling techniques and, as a consequence, the variance and the number of samples required for convergence are reduced. Also, we use a fractal dimension approach to characterize the error of the estimation of the recourse function based on sampling. The computational implementation of the algorithm involves a framework that integrates the GAMS modeling environment, the HSS sampling code (FORTRAN) and a C++ program which generates appropriate LP problems for each SD iteration. The algorithm has been tested with several case studies representing chemical engineering applications and the results are discussed.

Keywords: Stochastic programming; Stochastic decomposition; Hammersley sequence sampling technique

1. Introduction

There is a huge body of literature on stochastic linear programming including surveys and numerous articles (e.g., Birge & Louveaux, 1997; Bouza, 1993; Haneveld & vander Vlerk, 1999). The main class of stochastic linear problems involves two stages and the concept of recourse (SLPwR). In the first stage, the choice of the decision variable x is made. In the second stage, following the observation of the stochastic variables 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 Eqs. (1) and (2). Eq. (1) is referred as the first stage problem:

$$\min c^{\mathrm{T}}x + Q(x), \quad \text{s.t.} \quad Ax = b, \quad x \geq 0 \tag{1}$$

where A is a coefficient matrix, c a coefficient vector, and $Q(x)$ the recourse function defined by $Q(x) = E_u[Q(x, u)]$, where $Q(x, u)$ is obtained from the second stage problem (Eq. (2)):

$$Q(x, u) = \min q^{\mathrm{T}}(u)v, \quad \text{s.t.} \quad W(u)v = h(u) - T(u)x, \quad v \geq 0 \tag{2}$$

where q is a coefficient vector and W, h and T are the 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.

* Corresponding author. Tel.: +52-461-61-17575x156; fax: +52-461-61-17744.
E-mail address: vicente@iqcelaya.itc.mx (V. Rico-Ramirez).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.02.015

There is a rather complete optimization theory with necessary and sufficient conditions for SLPwR problems. However, they 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.

2. SLPwR algorithms

The two main algorithms for stochastic linear programming with fixed recourse are the L-shaped method (Birge & Louveaux, 1988, 1997) and the stochastic decomposition (SD) algorithm (Higle & Sen, 1991, 1996):

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.

2.1. 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. There are two types of cuts successively added during the solution procedure:

1. Feasibility cuts.
2. 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. 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 $Q(x)$. Rico-Ramirez (2002) provides a detailed description of the implications and derivation of both of the types of cuts.

2.2. 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. The algorithm presented here corresponds to the basic version provided by Higle and Sen (1996). Higle and Sen (1991, 1996) 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 the 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:

  $$\max \pi^{\mathrm{T}}(h_\nu - T_\nu x^\nu), \quad \text{s.t.} \quad \pi^{\mathrm{T}} W \leq q$$

  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:

  $$g_\nu^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^{\mathrm{T}} h_k$$

  $$G_\nu^\nu = \frac{1}{\nu}\sum_{k=1}^{\nu}(\pi_k^\nu)^{\mathrm{T}} T_k$$

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

  $$\max \pi^{\mathrm{T}}(h_k - T_k x^\nu), \quad \text{s.t.} \quad \pi \in V_\nu$$

  Observe that the solution vector to these 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:

  $$g_k^\nu = \frac{\nu - 1}{\nu} g_k^{\nu-1}, \quad k = 1, \ldots, \nu - 1$$

  $$G_k^\nu = \frac{\nu - 1}{\nu} G_k^{\nu-1}, \quad k = 1, \ldots, \nu - 1$$

- Step 3: Solve:

  $$\min c^{\mathrm{T}} x + \theta^\nu, \quad \text{s.t.} \quad Ax = b, \quad \theta^\nu + G_k^\nu x \geq g_k^\nu,$$

  $$k = 1, \ldots, \nu$$

  to obtain $x^{\nu+1}$. Go to step 1.
- There are several stopping criteria for the algorithm, such as those based on error bound estimates and optimality conditions. Here, however, only the most simple of the stopping criteria is considered. The algorithm stops if:
  (a) Change in the objective function is small.
  (b) No new dual vectors are added to the set of dual solutions, V.

Later in the paper, we will describe a computer implementation of this algorithm. It involves a framework that integrates the GAMS modeling environment, the HSS sampling

code and a C++ program which generates an LP problem for each SD iteration.

3. Hammersley sequence sampling

One of the most widely used techniques for sampling from a probability distribution is the Monte Carlo (MC) sampling technique. 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 (James, 1985). Importance sampling, Latin hypercube sampling (LHS) (Iman & Conover, 1982; Mckay, Beckman, & Conover, 1979) and descriptive sampling (Saliby, 1990) are examples of variance reduction techniques.

Recently, an efficient sampling technique (Hammersley sequence sampling) has been developed by Kalgnanam and Diwekar (1997), 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. Figs. 1 and 2 compare the result of using HSS and Monte Carlo to sample 100 points on a unit square (uniform distribution). The paper by Kalgnanam and Diwekar (1997) 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 3–100 times faster than LHS and Monte Carlo techniques and hence is a preferred technique for uncertainty analysis as well as optimization under uncertainty. The HSS sampling technique will be used in our implementation of the SD algorithm.

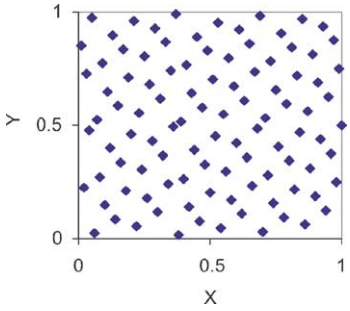

Fig. 1. Uniformity in Hammersley sequence sampling.

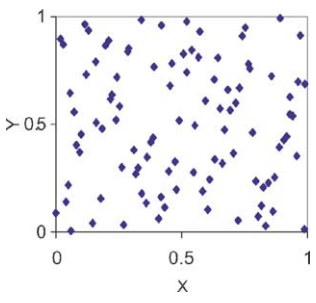

Fig. 2. Monte Carlo sampling.

3.1. Error estimation

The error bandwidth of samples generated by Monte Carlo sampling can be estimated from classic statistical methods. No matter what the distribution of u, the central limit theorem allows one to calculate the probabilistic error bands on the expected value of the random samples. For 95% confidence intervals, the error bandwidth of Monte Carlo sampling becomes:

$$\varepsilon_{\mathrm{MCS}} \propto \frac{1}{\sqrt{N}} \tag{3}$$

where N is the number of samples. The use of the HSS presents an implication from the point of view of statistical inference, since the central limit theorem cannot be applied to HSS samples. For that reason, Chaudhuri and Diwekar (1999) proposed a fractal dimension approach to characterize and predict the error when generating a large number of samples of the uncertain variable. Hence, for instance, in the case of stochastic annealing the following expression was derived:

$$\varepsilon_{\mathrm{HSS}} \propto \frac{1}{N^{1.8}} \tag{4}$$

A fractal approach will be used in this work in order to characterize the error of the approximations to the recourse function.

4. Implementation

The algorithm presented in Section 2.2 corresponds to the basic SD algorithm of Higle and Sen (1996). Such authors also suggest additional concepts to improve performance of the algorithm. For instance, they present a stabilization strategy of the algorithm based on incumbent solutions; furthermore, they prove the advantages of adding a regularizing term to the objective function which allows the elimination of redundant cuts. In this work, however, rather than analyzing the convergence properties, we are interested in the impact that an efficient sampling technique may have in the convergence of the algorithm. For that reason, only the basic SD algorithm will be used.

Hence, we compare the performance of the SD algorithm when the HSS and the MC sampling techniques are used to sample the random variables. The computational implementation of the algorithm involves a framework that integrates the GAMS modeling environment (Brooke, Kendrick, Meeraus, & Raman, 1999), the HSS sampling code (FORTRAN) and a C++ program which generates the appropriate LP problems for each SD iteration. The implementation

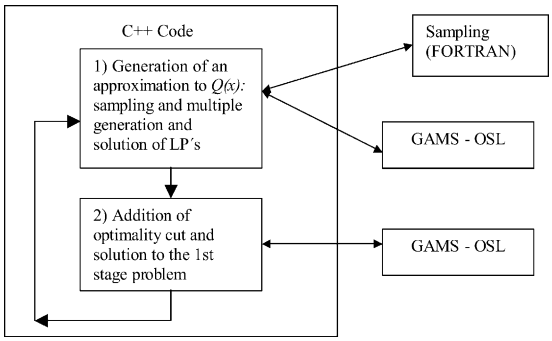

Fig. 3. Computational implementation of the algorithm.

is shown in Fig. 3. The master program is a C++ code which is in charge of calling GAMS (hundred of calls) and the sampling technique. After a realization of the random variable is obtained, the solution of the dual of the second stage problem is solved through GAMS-OSL. Then, the new cut is obtained and previous cuts are updated. The first stage problem is then solved and the algorithm starts over again with the sampling.

5. Application to chemical engineering examples

The stochastic version of three chemical engineering examples are used as case studies for evaluating the impact of using the HSS technique in the SD algorithm. These examples correspond to the stochastic versions of the examples presented by Edgar, Himmelblau, and Lasdon (2001). Table 1 provides the information about the basic structure of each of the three cases.

Table 1
Chemical engineering stochastic linear programming applications

Case study	First stage		Second stage		Random variables
	Rows	Columns	Rows	Columns	
Boiler/turbo generator system	1	2	21	28	4
Refinery planning	4	5	12	13	8
Petrochemical plant planning	1	2	11	15	11

5.1. Case study 1

The first case study is a boiler/turbo generator system problem. To produce electric power, this system contains two turbo generators (see Fig. 4). Turbine 1 is a double extraction turbine with two intermediate streams leaving at 195 and 62 psig; the final stage produces condensate that is used as boiled feed water. Turbine 2 is a single extraction turbine with one intermediate stream at 195 psig and an exit stream leaving at 62 psig with no condensate being formed. To meet the electric power demand, electric power must be purchased with a minimum base. The system may be modeled as linear constraints and a linear objective function. The demands on the resources are considered as uncertain variables in the problem.

5.2. Case study 2

The second problem is a refinery blending and production problem. There are four types of crude processed by the refinery. Also, the multi-unit refinery is disaggregated into

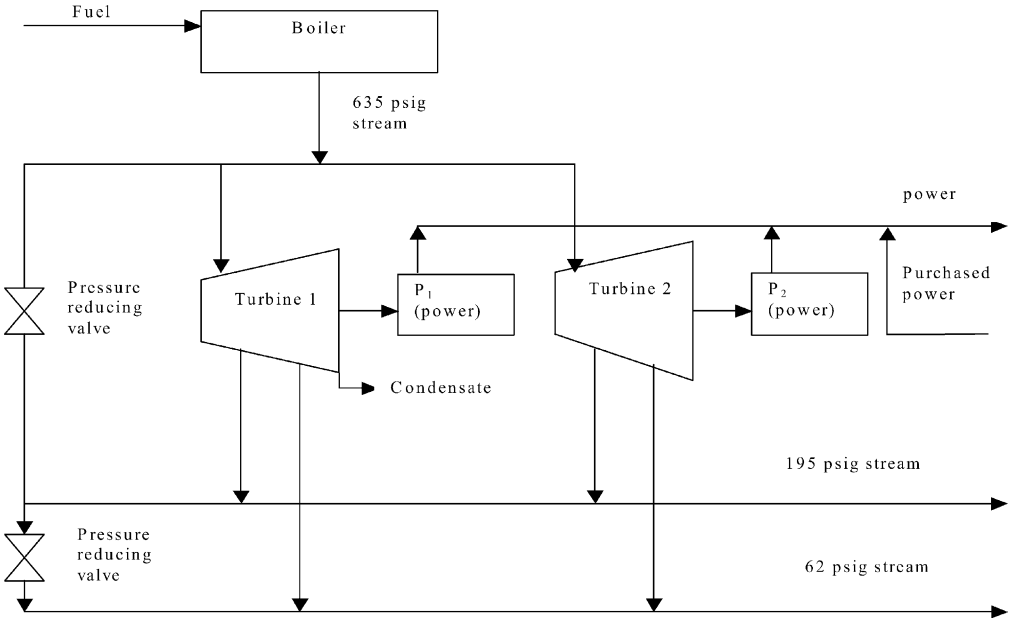

Fig. 4. Case study 1: boiler/turbo generator system.

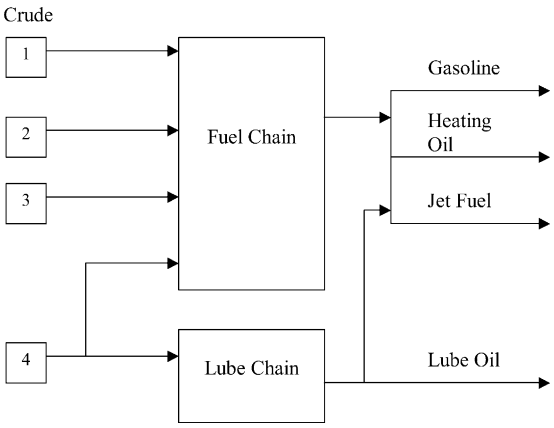

Fig. 5. Case study 2: refinery planning.

two processes: a fuel chain and a lube chain (see Fig. 5). The problem is to optimally allocate the crude between the two processes subject to supply and demand constraints.

5.3. Case study 3

The third problem is very similar to the previous one. A petroleum company receives five types of crude and obtain six different products. The problem consists on deciding the amount of crude which will be obtained from each source in order to satisfy the demands and product specifications.

5.4. Results

The values of the objective function as well as the error are presented for each of the three cases (see Figs. 6–11). Also, Table 2 shows a summary of the results obtained for each case. VSS stands for value of stochastic solu-

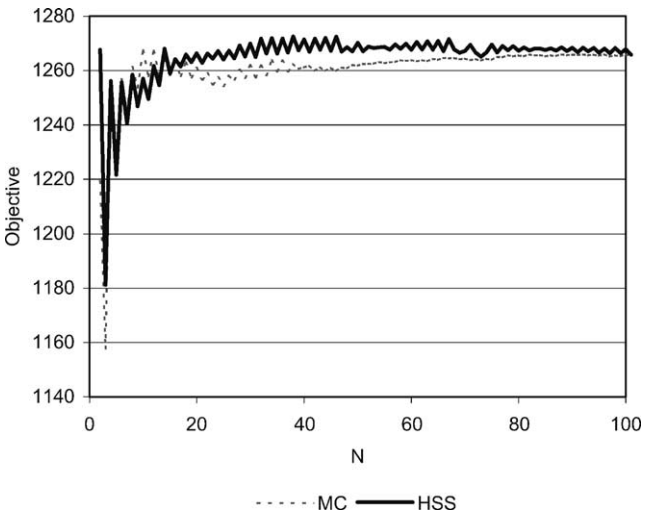

Fig. 6. Objective function for case study 1.

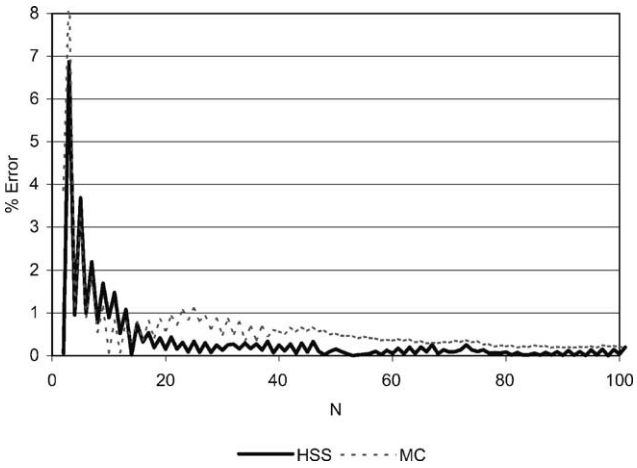

Fig. 7. Error for case study 1.

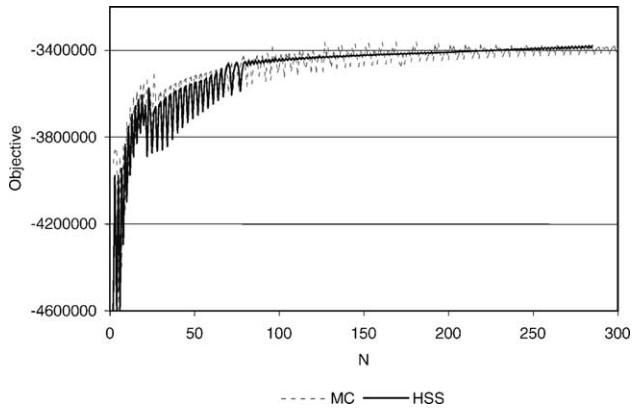

Fig. 8. Objective function for case study 2.

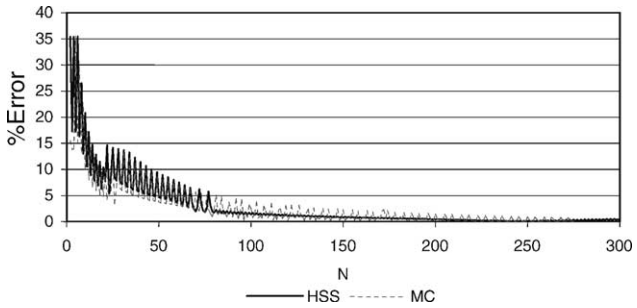

Fig. 9. Error for case study 2.

Table 2
Results

Case study	*N* (same average error)		Exponent *H* (HSS)	VSS (% objective)
	HSS	MC		
Boiler/turbo generator system	60	170	1.36	0.48
Refinery planning	190	275	1.86	6.85
Petrochemical plant planning	75	160	1.12	2.22

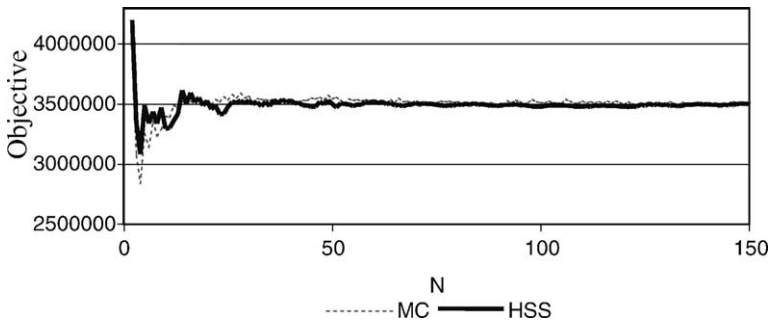

Fig. 10. Objective function for case study 3.

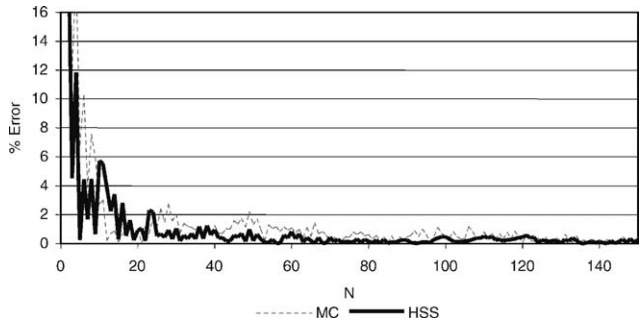

Fig. 11. Error for case study 3.

tion and is given as a percentage of the value of the objective function. H is the exponent used to characterize the error when using HSS for sampling. The first two columns try to show a comparison about the number of iterations needed in each of the methods (HSS and Monte Carlo) in order to get the same average error. It is evident that this number changes with the selected average value.

6. Discussion

For the three cases, it can be observed that the error presented with the HSS sampling is lower than that obtained with the MC sampling. Hence, the reduction on the number of iterations and the error seems to be a general advantage of HSS with respect to other sampling techniques. On the other hand, the fractal approach to characterize the error did not produce a consistent result as expected. The coefficients found for the HSS technique on the three cases are different from each other. The reason for that might be that the number of iterations used is relatively small for this type of analysis, since the expected result is supposed to apply for large values of N. Such a topic is now under investigation. Finally, the values of VSS shown in Table 2 indicate that the benefit of solving the problems as stochastic programs (instead of mean value problems) is very important in cases 2 and 3 but, for case 1, is not really significant.

7. Mixed-integer stochastic linear problems: preliminary analysis

Current research efforts focus on extending the proposed analysis for stochastic mixed-integer linear programs. It is expected that, since every node of the branch and bound algorithm might be individually seen as a SLP, the number of iterations and computer time when using HSS should be dramatically reduced. The numbers presented in Figs. 12 and 13 correspond to a very simple problem on which there are three binary variables, two deterministic continuous variables and two uncertain variables. For solving the problem, a simplistic approach was used. That it, it was considered

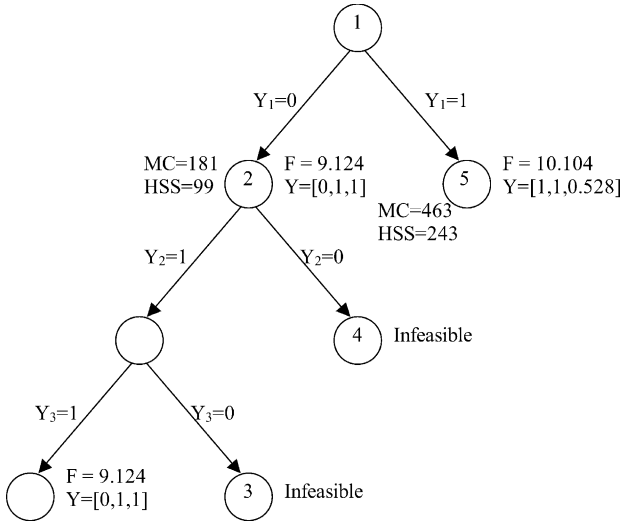

Fig. 12. Preliminary analysis in MISLP: depth first.

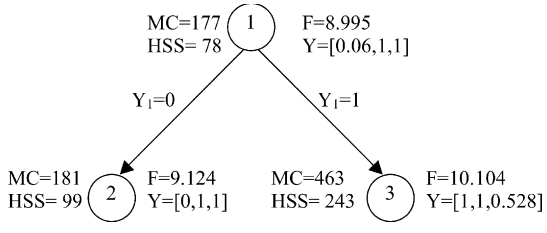

Fig. 13. Preliminary analysis in MISLP: breadth first.

that each node is a SLPwR problem and the total number of iterations can be compared when using HSS and MC. From looking at the numbers shown in Figs. 12 and 13, even for this simple example, we can observe a reduction of about 50% in the number of iterations needed.

8. Summary

Future cannot be perfectly forecasted but instead should be considered random or uncertain. The optimization under uncertainty is popularly known as stochastic programming. The need for including uncertainty in complex decision models of process systems engineering has been considered for decades but have gained renewed interest in recent years. In chemical engineering processes, qualitatively different sources of uncertainty may be located. For instance, uncertainty in the input variables (originated from the random nature and unpredictability of certain process inputs) and uncertainty with respect to the model parameters (resulting from the impossibility of modeling exactly the physical behavior of the system). Hence, stochastic programming problems appear in every field and in a wide variety of contexts and certainly chemical engineering processes are no exception.

This paper deals with stochastic linear programming problems and, in particular, with an approach to the solution of stochastic linear programming problems by using the SD algorithm. Although the application to just linear problems is indeed a limitation, there are several important application of linear programming in process systems engineering, such as scheduling and production planning models.

This research was not intended to test the numerical properties of the SD algorithm. The main purpose of the approach was to analyze the impact of using an efficient sampling technique embedded in the optimization algorithm for SLP problems. As a general result, one can conclude that the reduction on the number of iterations needed for convergence of the SD algorithm can be greatly reduced by using HSS instead of other sampling techniques. A preliminary analysis on MISLP problems also reveals the potential benefit of the approach on those cases. Finally, a meaningful conclusion about the effectiveness of the fractal dimension approach to characterize the error could not be drawn, since the number of samples required for convergence was relatively small for each of the problems considered. Such an analysis is the focus of ongoing research.

Acknowledgements

Jose M. Ponce-Ortega and Vicente Rico-Ramirez would like to thank the financial support given by “Consejo Nacional de Ciencia y Tecnologia, CONACYT” and by the “Consejo del Sistema Nacional de Educacion Tecnologica, CoSNET”.

References

Birge, J. R., & Louveaux, F. (1988). A multicut algorithm for two-stage stochastic linear programming. *European Journal of Operations Research, 34*, 384–392.

Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. New York: Springer-Verlag.

Bouza, C. (1993). Stochastic programming: The state of the art. *Revista Investigacion Operacional, 14*(2) 148–161.

Brooke, A., Kendrick, D., Meeraus, A., & Raman, R. (1999). *GAMS—A user guide*. Washington, DC: GAMS Development Corporation.

Chaudhuri, P., & Diwekar, U. M. (1999). Synthesis approach to the determination of optimal waste blends under uncertainty. *AIChE Journal, 45*(8), 1671–1687.

Edgar, T. F., Himmelblau, D. M., & Lasdon, L. S. (2001). *Optimization of chemical processes*. New York: McGraw-Hill.

Haneveld, W. K. K., & vander Vlerk, M. H. (1999). Stochastic integer programming: General models and algorithms. *Annals of Operational Research, 85*, 39–57.

Higle, J. L., & Sen, S. (1991). Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research, 16*(3), 650–669.

Higle, J. L., & Sen, S. (1996). *Stochastic decomposition*. Kluwer Academic Publishers.

Iman, R. L., & Conover, W. J. (1982). Small sample sensitivity analysis techniques for computer models, with an application to risk assessment. *Communications in Statistics, A17*, 1749.

James, B. A. P. (1985). Variance reduction techniques. *Journal of the Operations Research Society, 36*(6), 525.

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

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

Rico-Ramirez, V. (2002). Two-stage stochastic linear programming: A tutorial. *SIAG/OPT Views and News, 13*(1), 8–14.

Saliby, E. (1990). Descriptive sampling: A better approach to Monte Carlo simulations. *Journal of the Operations Research Society, 41*(12), 1133.