SAMPLING TECHNIQUES

1. Introduction

Sampling is a statistical procedure that involves the selection of a finite number of individuals to represent and infer some knowledge about a population of concern. Sampling techniques are used in a wide range of science and engineering applications; they are of basic importance in computational statistics, in the implementation of probabilistic algorithms, and in related problems of statistical computing that have a stochastic ingredient (eg, financial modeling, artificial intelligence, computational chemistry, risk and uncertainty analysis, and design of experiments). This article is devoted to the role of sampling in process systems engineering.

Uncertainty analysis is a crucial step in process design and development due to the fact that over the life cycle of the process, product demands change, there may be variations in feedstock and product specifications, and the process may be subject to short- and long-term uncertainties. Furthermore, increased environmental consciousness in recent years and the efforts for pollution prevention necessitate chemical manufacturing plants to comply with stricter environmental regulations and to reduce waste. Therefore, the breadth of traditional process design approaches should be extended to include green engineering principles early in design (1,2). The reason is because the decisions made earlier during the development of a chemical process affect later stages, eg, material and equipment selection, pilot plant studies, and financial analysis and because the opportunities for reducing environmental and health impacts of a process diminishes. Therefore, unlike traditional process design, where engineers are seeking only low cost options, contemporary process design approaches include environmental and health impacts, process performance indexes, eg, risk, reliability, safety, and flexibility, as well as controllability and profitability into decision making. Sampling plays an important role in defining and quantifying these objectives. Further, nowadays process design is just not restricted to process simulation, but includes steps, eg, discovery, chemical synthesis on one end, and management, planning, and control on the other end. As the breadth of this design framework is extended, uncertainties in the model increase and efficient algorithms and tools are needed to address this problem. Sampling is an important component of these algorithms and tools.

Figure 1 shows an overview of this integrated framework proposed (2), which applies green engineering principles at every stage of process design and development. The first stage in process development is discovery, where chemicals and materials are selected and synthesized in a laboratory or using computational chemistry methods. These methods use Monte Carlo methods based on sampling of the molecular configurational space.

Computer-aided molecular design (CAMD) is a commonly used technique for chemical synthesis where the reverse use of group contribution methods is employed to select materials with desired physical, chemical, environmental, and biological properties. The next stage of chemical synthesis is process synthesis, where a chemical process is developed by choosing various unit operations and their connections. A flowsheet of the proposed plant is generated and process simulators are used to compute mass and energy flows for the process to predict its behavior if it was constructed.

Uncertainties are commonly present in chemical and process synthesis due to insufficient experimental data and the lack of accurate models for representing the physical and chemical phenomena. Uncertainties are also encountered over the life cycle of the plant that affect decisions related to plant operations, eg, process control, production planning and scheduling, supply chain management, reliability, and maintenance of the plant.

For example, model uncertainty and external disturbances are important concerns in designing control systems, which are used to minimize deviations from the nominal process conditions and maintaining the safe operation of the plant. Probabilistic approaches and sampling techniques are used commonly to ensure robustness to these model uncertainties. On the other hand, off-line

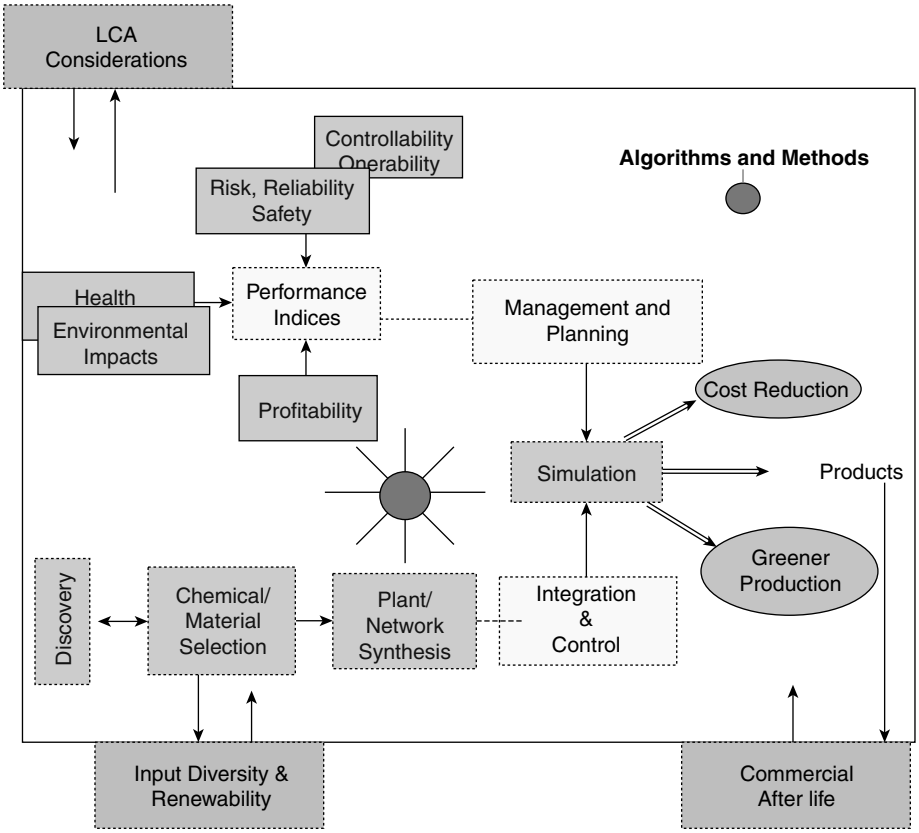

Fig. 1. Integrated framework for environmentally conscious process development and design under uncertainty (2).

quality control is used to design products and processes that are robust to uncontrollable variation at the design stage. Parameter design strategy is used for this purpose and sampling techniques are employed to propagate the effects of input variability on outputs. The choice of an efficient sampling technique is very important for efficient off-line quality control.

For multipurpose/multiproduct batch plants, optimal production planning and scheduling is important in order to be competitive in a just-in-time production environment. The scheduling problem assigns a sequence of tasks to each equipment over time, according to inventory restrictions and customer demands. The production schedule should be able to accommodate changing product demands, equipment shutdowns and unexpected orders. An extension of the scheduling problem is supply chain management that deals with a complex network of suppliers, plant, warehouses, distribution centers, and customers. Examples of uncertainties in supply chains include fluctuations in product prices, demands, or production yields.

In order to increase operational effectiveness and profits, and to save on lost production and costs, chemical plants need to operate with high process reliability and availability. Therefore, reliability issues need to be addressed at the conceptual

design stage. Optimal maintenance schedules for the plant need to be determined to increase reliability and availability while maintaining profitability. Uncertainties in equipment availability profoundly affect the profitability of the plant.

Uncertainty analysis and sampling techniques also play an important role in risk assessment and safety. Risk management is a decision making process that is used to reduce the financial and production risk for a business. Environmental risk assessment and financial risk assessment are commonly applied to chemical manufacturing processes. Environmental risk is associated with the toxicity of materials and the effect of hazardous materials on a human population or an entire ecosystem. Financial risk on the other hand, is concerned with pricing decisions and demands. Probability distributions and sampling techniques are frequently used in risk and policy analysis.

The most commonly used sampling technique for uncertainty analysis is Monte Carlo sampling, which is based on a pseudo-random number generator. This sampling technique has probabilistic error bounds and large sample sizes are needed to achieve the desired accuracy. Variance reduction techniques have been applied to circumvent the disadvantages of Monte Carlo sampling.

2. Sampling Techniques

Sampling is a statistical procedure that involves selecting a limited number of observations, states, or individuals from a population of interest. A sample is assumed to be representative of the whole population to which it belongs. Instead of evaluating all the members of the population, which would be time consuming and costly, sampling techniques are used to infer some knowledge about the population.

Sampling techniques could be divided into two groups: *probability sampling* and *nonprobability sampling*. In probability sampling, samples are selected based on the theory of probability, which means that each possible set of unit is assigned a probability of selection. The samples are selected by a random process and the confidence intervals for the estimates are known. On the other hand, nonprobability sampling does not involve random selection of individuals. An example of this is quota sampling, where the population is first divided into subpopulations and subjects are selected according to judgment or convenience. In this case, the sampling error cannot be determined by probabilistic techniques.

For a good sampling technique, all physically reasonable values of the input and output variables should have some chance of occurring and no region of the population should be excluded. Furthermore, the estimates should be as close as possible to the real values of the quantities being estimated. A good sampling technique also allows an assessment of the relative importance of each input variable.

Probabilistic sampling techniques are based on Monte Carlo methods and are most relevant to this article. They are described in this section.

2.1. Monte Carlo Sampling. One of the simplest and most widely used methods for sampling is the Monte Carlo method. Monte Carlo methods are numerical methods that provide approximate solutions to a variety of physical and mathematical problems by random sampling. The name Monte Carlo, which was suggested by Nicholas Metropolis, takes its name from a city in the

Monaco principality, which is famous for its casinos, because of the similarity between statistical experiments and the random nature of the games of chance, eg, roulette.

Monte Carlo methods were originally developed for the Manhattan Project during World War II to simulate probabilistic problems related random neutron diffusion in fissile material. Although they were limited by the computational tools of that time, they became widely used in many branches of science after the electronic computers were built in 1945. The first publication that presents the Monte Carlo algorithm is probably by Metropolis and Ulam (3).

The basic idea behind Monte Carlo simulation has been that input samples should be randomly generated in order to describe a random output. In the crude Monte Carlo approach, a value is drawn at random from the probability distribution for each input, and the corresponding output value is computed. The entire process is repeated n times producing n corresponding output values. These output values constitute a random sample from the probability distribution over the output induced by the probability distributions over the inputs. The simplest distribution that is approximated by the Monte Carlo method is a uniform distribution U(0,1) with n samples on a k-dimensional unit hypercube. One advantage of this approach is that the precision of the output distribution may be estimated using standard statistical techniques. On average, the error ε of approximation is of the order $O(N^{-1/2})$. One remarkable feature of this sampling technique is that the error bound is not dependent on the dimension k. However, this bound is probabilistic, which means that there is never any guarantee that the expected accuracy will be achieved in a concrete calculation.

The success of a Monte Carlo calculation depends on the choice of an appropriate random sample. The required random numbers and vectors are generated by the computer in a deterministic algorithm. Therefore, these numbers are called *pseudorandom numbers* or *pseudorandom vectors*. One of the oldest and best known methods for generating pseudorandom numbers for Monte Carlo sampling is the liner congruential generator (LCG), first introduced by Lehmer (4). The general formula for a linear congruential generator follows:

$$I_n = (a \cdot I_{n-1} + c) \bmod m \tag{1}$$

In this formula, a is the multiplier, c is the increment that is typically set to zero, and m is the modulus. These are preselected constants. The proper choice of these constants is very important for obtaining a sample that performs well in statistical tests. One other preselected constant is the *seed* I_0, which is the first number in the output of a linear congruential generator. The following example shows how to generate pseudorandom numbers.

Example 1: An example of a library that generates pseudorandom numbers combines three generators (5):

$$\begin{aligned} X_n &= (171 \cdot X_{n-1}) \bmod 30269 \\ Y_n &= (172 \cdot Y_{n-1}) \bmod 30307 \\ Z_n &= (170 \cdot Z_{n-1}) \bmod 30323 \end{aligned} \tag{2}$$

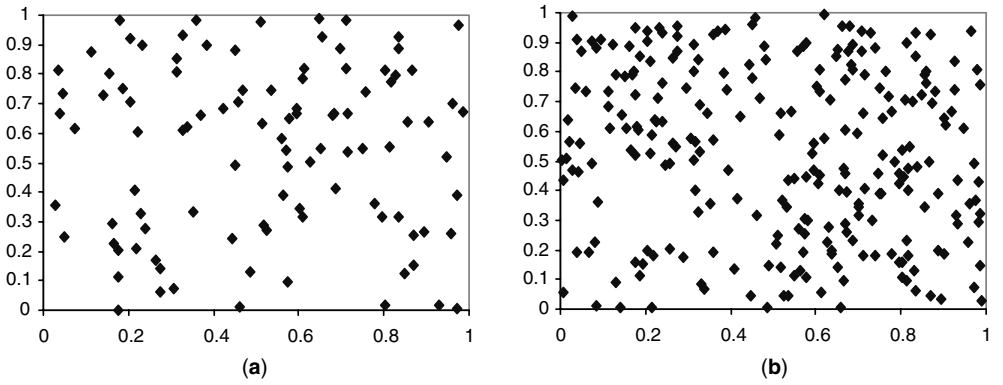

Fig. 2. (**a**) 100 pseudorandom numbers on a unit square, (**b**) 250 pseudorandom numbers on a unit square obtained by the linear congruential generator developed by Wichmann and Hill (5).

The random number is calculated from the formula:

$$\text{temp} = X_n/30629.0 + Y_n/30307.0 + Z_n/30323.0 \tag{3}$$

$$\text{random} = \text{temp} - \text{int}(\text{temp}) \tag{4}$$

Pseudorandom numbers of different sample sizes on a unit square generated using this method is given in Figure 2. From this figure, it can be seen that pseudorandom number generator produces samples that may be clustered in certain regions of the unit square and does not produce uniform samples. Therefore, in order to reach high accuracy, larger sample sizes are needed, which adversely affects the efficiency of this method.

Monte Carlo method provides approximate solutions to a variety of mathematical problems. A classic use of Monte Carlo methods is for the evaluation of definite integrals, particularly multidimensional integrals with complicated boundary conditions. A simple example for the estimation of π and how the sample size affects the accuracy of estimation is given below in Example 2.

$$\frac{\text{Area}_{(\text{circle})}}{\text{Area}_{(\text{square})}} \times 4 = \frac{\pi \cdot r^2}{4 \cdot r^2} \times 4 = \pi$$

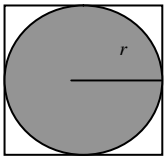

Fig. 3. A procedure for the estimation of π.

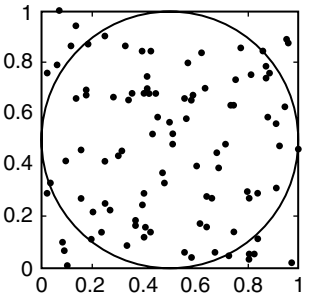

Fig. 4. Estimation of π by Monte Carlo sampling.

Example 2: The value of π could be obtained by dividing the area bounded by the circle to the area bounded by the square shown in Figure 3 and multiplying this value by 4.

The value of π can be estimated by Monte Carlo sampling. Two uniform random samples $U(0,1)$ could be generated to place on a unit square. The number of sample points bounded by the circle could be divided by the number of points bounded by the square and multiplied by 4 to obtain the estimation.

In Figure 4, an illustration of this procedure is shown for 100 Monte Carlo samples. The value of π estimated by these 100 samples is 3.04. As the sample size increases, the estimations come closer to the actual value of π, which is 3.1415 as shown in Figure 5.

2.2. Variance Reduction Techniques. For increasing the efficiency of Monte Carlo simulations and overcome the disadvantages, eg, probabilistic error bounds, variance reduction techniques have been developed.

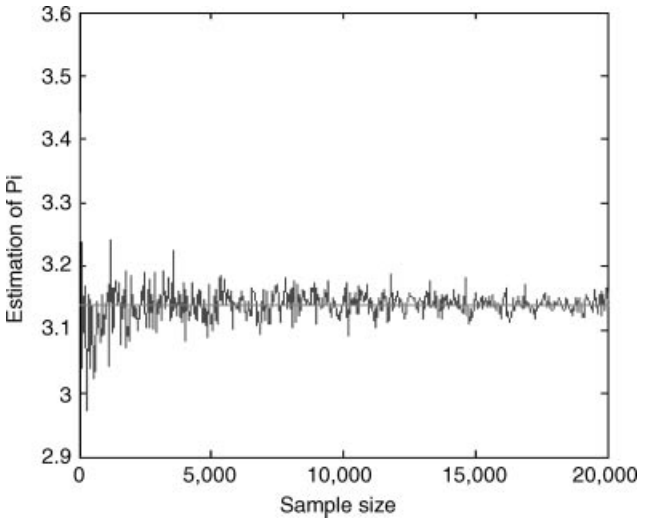

Fig. 5. Effect of sample size on the estimation of π.

James (6) has divided variance reduction techniques into four categories. However, note that these categories are not all inclusive. Further, some of the sampling techniques are overlapping between categories.

The first category of sampling techniques extract from a run more information than immediately evident on the parameter value. An example of this is the *control variate sampling*, which is one of the most versatile variance reduction techniques. Control variates exploit information about the errors in estimates of known quantities to reduce the error in an estimate of an unknown quantity (7). They are best able to estimate the mean of the outcome distribution, but also can help in variance estimation. Control variates are set up in order to use a simplified version of a model. One of the problems with control variates is the selection of effective controls. Also, control variates assume a specific probabilistic structure for the simulation output process, usually joint normality of the response and the control variates, and this underlying assumption may not always be satisfied (8). These problems restrict the widespread use of control variates. Therefore, it is not surprising that this method is not percolated in process systems engineering.

Sampling techniques in the second category make sure that each individual run is unbiased with respect to the mean outcome measure being estimated. For example, in *antithetic sampling*, a negative correlation is introduced between two unbiased estimators of a variable X (9). This technique is applied if there is only one important variable within the model, which is sampled once during a run. Similar to control variate sampling, so far we have not seen any application of this method in chemical engineering literature.

The sampling approaches for variance reduction that are used more frequently for chemical engineering applications are importance sampling, Latin Hypercube Sampling (LHS) (10,11), Descriptive Sampling (12), and Hammersley Sequence Sampling (HSS) (13,14). The latter technique belongs to the group of quasi-Monte Carlo methods that were introduced in order to improve the efficiency of Monte Carlo methods by using quasirandom sequences that show better statistical properties and deterministic error bounds. These commonly used sampling techniques are described below with examples.

Importance Sampling. Importance sampling, which may also be called *biased* sampling, is a variance reduction technique for increasing the efficiency of Monte Carlo algorithms. Monte Carlo methods are commonly used to integrate a function F over the domain D:

$$\int_D F(x)dx \tag{5}$$

If random numbers are drawn from a normal distribution, information is spread over the interval being sampled. However, if a nonuniform (biased) distribution $G(x)$ is used, which draws more samples from the areas that make a substantial contribution to the integral, the approximation of the integral will be more accurate, and the process will be more efficient. This is the basic idea behind importance sampling, where the approximated integral is given by

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{F(x_i)}{G(x_i)} \tag{6}$$

Importance sampling is crucial for sampling low probability events. The most critical issue for the implementation of importance sampling is the choice of the biased distribution that emphasizes the important regions of the input variables. One of the examples of importance sampling is the Metropolis criterion used in molecular simulations (15). In molecular simulations the configurational phase space is explored and this involves the evaluation of a multidimensional integral over $3N$ degrees of freedom. The crucial feature of the Metropolis approach is that it generates a Markov chain of states and it biases the generation of configurations toward those that make the most significant contribution to the integral. Specifically, it generates states with a probability exp $(-\Delta V/k_{\mathrm{B}}T)$, where ΔV is the change in energy, k_{B} is the Boltzmann factor, and T is the temperature. This algorithm allows the low energy configurations to be sampled more efficiently, where the Boltzmann factor has an appreciable value. As a result, thermodynamic properties of fluids could be calculated more accurately. A simple example for the application of importance sampling for estimation of a simple integral is given below.

Example 3: Let us assume that one would like to estimate the integral:

$$I = \int_0^\infty x^2 \exp^{-x^2} dx \tag{7}$$

This function is not possible to integrate analytically, but its value is known to be $\sqrt{\pi}/4 = 0.44311328....$

As seen from Figure 6, the value of this function decreases rapidly when x is greater than $\sim$3.5. Therefore, there are only a small number of input arguments, x, where the integral has an appreciable value.

If a Monte Carlo integration is applied to estimate this integral, the domain of this integral can be uniformly sampled by using a uniform distribution between 0 and 1000. Then this integral can be evaluated using the uniform intervals.

However, it is known that this integral has only an appreciable value at a specific interval. Because of that, if a uniform sample is used, most of the points

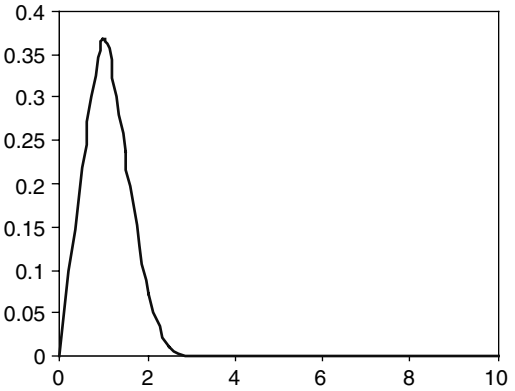

Fig. 6. The function $f(x) = x^2 \exp(-x^2)$.

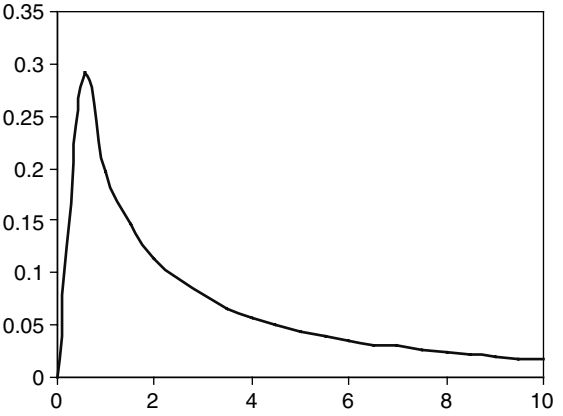

Fig. 7. Lognormal distribution with a mean $\mu = 1$ and a standard deviation of $\sigma = 1.7$.

will be from areas that correspond to values where the integral has a very small value. Therefore, a nonuniform distribution function can be used instead for sampling. If a distribution like log-normal distribution is chosen, the number of samples required to obtain an accurate estimation will be less. For example, consider a lognormal distribution with mean $\mu = 1$ and a standard deviation of $\sigma = 1.7$. This is shown in Figure 7. It is seen that, if a lognormal distribution is used, there will be more sampling from the areas of importance that make a significant contribution to the integral.

The estimation of this integral using a uniform sample and a lognormal sample is compared in Table 1. The integral is accurately estimated using importance sampling after only 100 samples. However, it requires 10,000 samples with the crude Monte Carlo method where a uniform distribution is used.

Stratified Sampling. Stratification is the grouping of the members of a population into equal or unequal probability areas (strata) before sampling. The strata must be mutually exclusive, which means that every element in the population must be assigned to only one stratum. Also, no population element is excluded. It is required that the proportion of each stratum in the sample should be the same as in the population.

Latin hypercube sampling is one form of stratified sampling that can yield more precise estimates of the distribution function (10), and therefore reduce the number of samples required to improve computational efficiency. It is a full

Table 1. **The Estimation of the Integral $f(x) = x^2 \exp(-x^2)$By Using Uniform Random Sampling and Importance Sampling**

n	Uniform random sampling	Importance sampling
10	0	0.11054
100	0.00095	0.44363
1,000	0.07585	0.44312
10,000	0.44131	0.44311

stratification of the sampled distribution with a random selection inside each stratum. In LHS, the range of each uncertain parameter X_i is subdivided into nonoverlapping 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 (ie, 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, etc, until n k-tuplets are formed.

An example is given to help clarify how intervals are formed.

Example 4: Consider the generation of a LHS of size $n = 5$ with two input variables. Let us assume that the first random variable X_1 has a normal distribution with a mean value of $\mu = 8$ and a standard deviation of $\sigma = 1$. The endpoints of the intervals are easily determined based on the parameters μ and σ^2. The intervals are shown in Figure 8 on both the density function and the cumulative distribution function. Each interval corresponds to a 20% probability.

It is also assumed that, the second random variable X_2 has a uniform distribution on the interval from 5 to 10. Therefore, one can easily determine the corresponding intervals in terms of both the density function and the cumulative distribution function as shown in Figure 9.

The next step is to obtain a LHS to choose specific values of X_1 and X_2 in each of their five respective intervals. This selection is done in a random manner with respect to density in each interval. Next, the selected values of X_1 and X_2

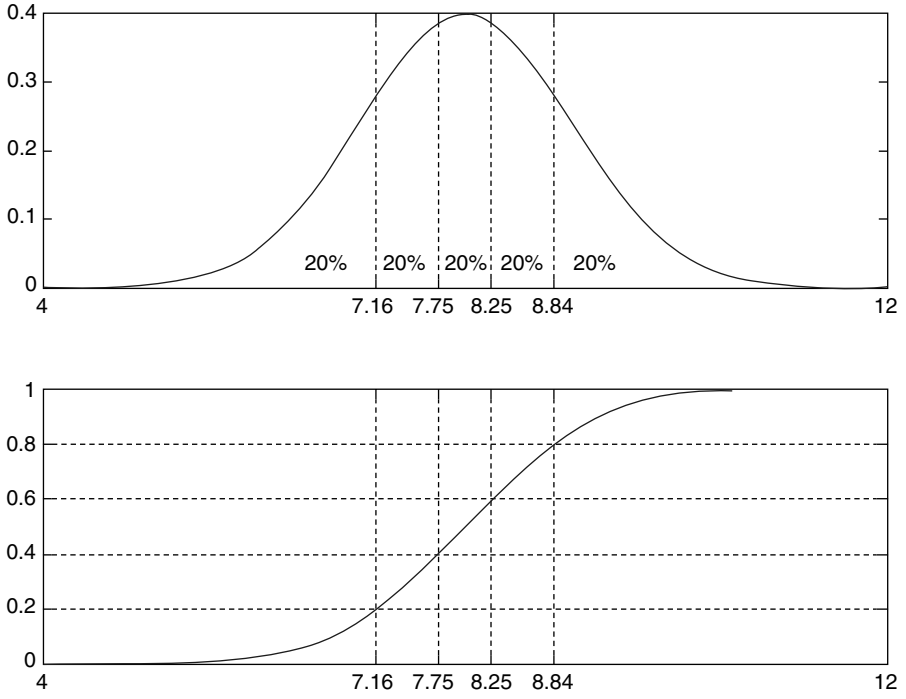

Fig. 8. Intervals used with a LHS of size $n = 5$ in terms of the density function and cumulative distribution function for a normal random variable X_1.

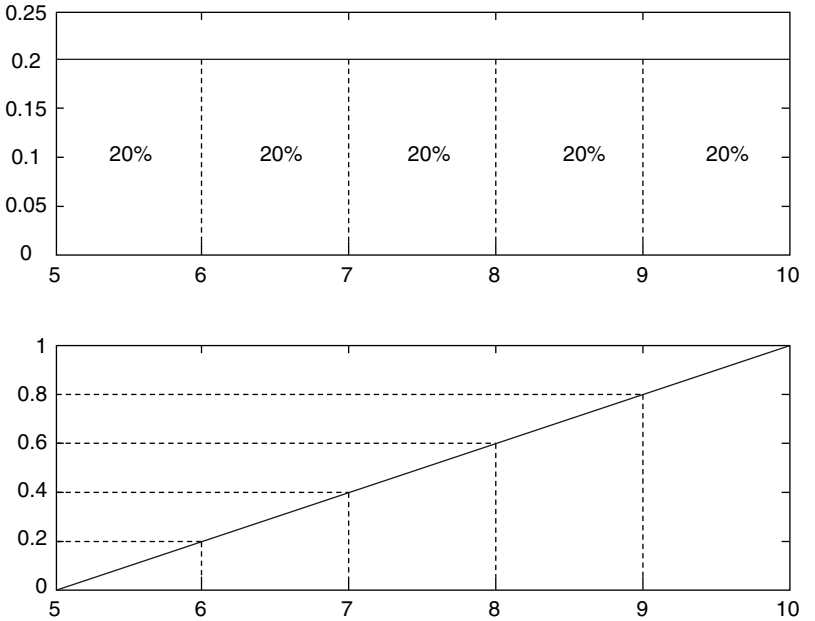

Fig. 9. Intervals used with a LHS of size $n = 5$ in terms of the density function and cumulative distribution function for the uniform random variable X_2.

are paired randomly to form the two-dimensional (2D) input vectors of size 5. This pairing is done by a random permutation of the first five integers with each input variable.

For example, two random permutations of the integers (1, 2, 3, 4, 5) can be considered:

$$\text{Permutation 1: } (2, 5, 3, 1, 4) \quad \text{Permutation 2: } (4, 3, 2, 5, 1)$$

These can be used as interval numbers for X_1 (Permutation 1) and X_2 (Permutation 2). In order to get the specific values of X_1 and X_2, $n = 5$ random numbers are randomly selected from the standard uniform distribution. If these values are denoted by U_m, where $m = 1, 2, 3, 4, 5$. Each random number U_m is scaled to obtain a cumulative probability P_m, so that each P_m lies within the mth interval:

$$P_m = \left(\frac{1}{5}\right) U_m + \left(\frac{m-1}{5}\right) \tag{8}$$

In Tables 2 and 3, possible selections of LHS of size 5 for random variables X_1 and X_2 are presented, respectively. Therefore, if the two permutations (Permutation 1 and 2) are applied to choose the corresponding intervals for X_1 and X_2, as given in Table 4, the pairing operation can be performed. This pairing process is illustrated in Figure 10. From this figure, it can be seen that all the intervals of X_1 and X_2 have been sampled.

Table 2. **Possible Selection of Values for a LHS of Size 5 for the Random Variable X_1, Normally Distributed with Mean $\mu = 5$ and $\sigma = 1$**

Interval number, m	Uniform (0,1)	Scaled probabilities $P_m = U_m(0.2) + (m-1)^*(0.2)$	Corresponding $U(5,10)$
1	0.5832	0.1166	6.808
2	0.8125	0.3625	7.648
3	0.2980	0.4596	7.899
4	0.8470	0.7694	8.737
5	0.4369	0.8874	9.213

Table 3. **Possible Selection of Values for a LHS of Size 5 for the Random Variable X_2, Uniformly Distributed between 5 and 10**

Interval number, m	Uniform (0,1)	Scaled probabilities $P_m = U_m(0.2) + (m-1)^*(0.2)$	Corresponding $U(5,10)$
1	0.3370	0.0674	5.337
2	0.1678	0.2336	6.168
3	0.8419	0.5684	7.842
4	0.4372	0.6874	8.437
5	0.8127	0.9625	9.813

Latin hypercube sampling was designed to improve the uniformity properties of Monte Carlo methods, since it was shown that the error of approximating a distribution by finite sample depends on the equidistribution properties of the sample used for $U(0,1)$, and the relationship between successive points in a sample or its randomness or independence is not critical (16).

In median Latin hypercube sampling (MLHS), which is a variant of LHS, the midpoint of the intervals is chosen to sample the uncertain variables. This sampling is similar to the *Descriptive Sampling* described by Saliby (12).

The main drawback of this stratification scheme in LHS and MLHS is that it is uniform in one dimension (1D) and does not provide uniformity properties in k dimensions. Sampling based on quadrature (17), cubature techniques (18), or collocation techniques (19) face similar drawback. These sampling techniques

Table 4. **Application of Random Permutations to Choose Intervals for X_1 and X_2 for Pairing**

Permutation 1 (interval used for X_1)	X_1	Permutation 2 (interval used for X_2)	X_2
2	7.648	4	8.437
5	9.213	3	7.842
3	7.899	2	6.168
1	6.808	5	9.813
4	8.737	1	5.337

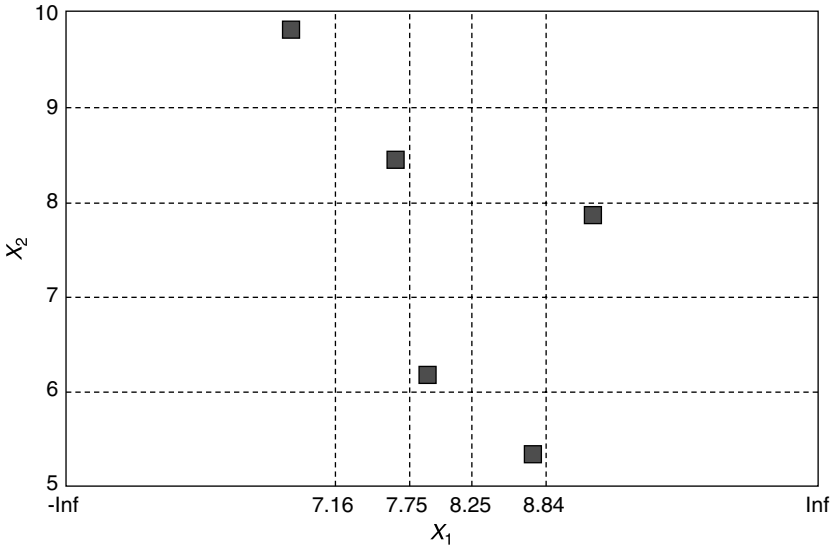

Fig. 10. Two-dimensional representation of a possible LHS of size 5 using X_1 and X_2.

perform better for lower dimensional uncertainties. Therefore many of these sampling techniques use correlations to transform the integral into 1D or 2D. However, this transformation is possible only for limited distribution functions when the uncertain variables are tightly correlated. For highly correlated samples, similar to what has been observed in thermodynamic phase equilibria, a sampling technique based on confidence region estimates can be used (20).

Quasi-Monte Carlo Methods. Quasi-Monte Carlo methods seek to construct a sequence of points that perform significantly better than Monte Carlo, which has an average case of complexity of the order of $1/\varepsilon^2$. For a suitably chosen set of samples, the quasi-Monte Carlo method provides a deterministic error bound of the order $N^{-1}(\log N)^{k-1}$ without any strong assumptions about the integrand. Some well-known quasi-Monte Carlo sequences are *Halton*, *Hammersley*, *Sobol*, *Faure*, *Korobov*, *and Neiderreiter* (21). The choice of an appropriate quasi-Monte Carlo sequence is a function of discrepancy. The deterministic upper and lower error bounds of any sequence for integration are expressed in terms of the discrepancy measure. Discrepancy is a quantitative measure for the deviation of the sequence from the uniform distribution. Therefore, it is desirable to choose a low discrepancy sequence. The Halton (22) and Hammersley (23) are some examples of low discrepancy sequences.

Hammersley sequence sampling is an efficient sampling technique developed by Diwekar and co-workers (13,14,24) based on quasirandom numbers. Hammersley sequence sampling uses Hammersley points to uniformly sample a unit hypercube and inverts these points over the joint cumulative probability distribution to provide a sample set for the variables of interest.

The design of Hammersley points is given below. Any integer n can be written in radix-R notation (R is an integer) as follows:

$$n \equiv n_m n_{m-1} \ldots n_2 n_1 n_0 \tag{9}$$

$$n = n_0 + n_1 R + n_2 R^2 + \ldots + n_m R^m \tag{10}$$

where $m = [\log_R n] = [\ln n / \ln R]$ (the square brackets denote the integral part). A unique fraction between 0 and 1 called the inverse radix number can be constructed by reversing the order of the digits of n around the decimal point as follows:

$$\varphi_R(n) = n_0 n_1 n_2 \ldots n_m = n_0 R^{-1} + n_1 R^{-2} + \ldots + n_m R^{-m-1} \tag{11}$$

The Hammersley points on a k-dimensional cube are given by the following sequence:

$$\vec{z}_k(n) = \left(\frac{n}{N}, \varphi_{R_1}(n), \varphi_{R_2}(n), \ldots, \varphi_{R_{k-1}}(n)\right) \quad n = 1, 2, \ldots, N \tag{12}$$

where $R_1, R_2, \ldots, R_{k-1}$ are the first $k-1$ prime numbers. The Hammersley points are $\vec{x}_k(n) = 1 - \vec{z}_k(n)$. Example 5 illustrates Hammersley points are generated.

Example 5: Two-dimensional Hammersley points are generated with a sample size of 100. In this case, $N = 100$ and $k = 2$. The procedure for generating Hammersley points is given below for the first 10 points in Table 5.

As shown in Figure 11, Hammersley sequence sampling technique uses an optimal design scheme for placing n points on a k-dimensional hypercube. This scheme ensures that it is more representative of the population showing uniformity properties in multidimensions, unlike Monte Carlo, Latin hypercube and its variant MLHS techniques. A qualitative picture of the uniformity properties of the different sampling techniques on a unit square is presented in Figure 12. It is clearly observed that HSS shows better uniformity than other stratified sampling techniques, eg, LHS, which are uniform along a single dimension only and do not guarantee a homogeneous distribution of points over the multivariate probability space.

One of the main advantages of Monte Carlo methods is that the number of samples required to obtain a given accuracy of estimates does not scale exponentially with the number of uncertain variables. The HSS preserves this property of Monte Carlo. For correlated samples, the approach used by Kalagnanam and Diwekar (13) uses rank correlations (11) to preserve stratified design along each dimension. Although this approach preserves the uniformity properties of the stratified schemes, the optimal location of the Hammersley points is perturbed by imposing the correlation structure. Figure 13 illustrates the effect of imposing a correlation structure on the sample sets. The HSS technique has

Table 5. **Generation of 10 Hammersley Points in 2D**

n	$\vec{z}_k(n)$	n_2 - radix	$\varphi_2(n)$ - inverse radix	$\vec{x}_k(n) = 1 - \vec{z}_k(n)$
0	$(0, \varphi_2(0))$	0	$0/2^1 = 0$	(1-0, 1-0)**(1.0, 1)**
1	$(0.01, \varphi_2(1))$	1	$1/2^1 = 0.5$	(1-0.01, 1-0.5) **(0.99, 0.5)**
2	(0.02, $\varphi_2(2)$)	10	$\frac{0}{2^1}+\frac{1}{2^2} = 0.25$	(1-0.02, 1-0.25) **(0.98, 0.75)**
3	(0.03, $\varphi_2(3)$)	11	$\frac{1}{2^1}+\frac{0}{2^2} = 0.75$	(1-0.03, 1-0.75) **(0.97, 0.25)**
4	(0.04, $\varphi_2(4)$)	100	$\frac{0}{2^1}+\frac{0}{2^2}+\frac{1}{2^3} = 0.125$	(1-0.04, 1-0.125) **(0.96, 0.875)**
5	(0.05, $\varphi_2(5)$)	101	$\frac{1}{2^1}+\frac{0}{2^2}+\frac{1}{2^3} = 0.625$	(1-0.05, 1-0.625) **(0.95, 0.375)**
6	(0.06, $\varphi_2(6)$)	110	$\frac{0}{2^1}+\frac{1}{2^2}+\frac{1}{2^3} = 0.375$	(1-0.06, 1-0.375) **(0.94, 0.625)**
7	(0.07, $\varphi_2(7)$)	111	$\frac{1}{2^1}+\frac{1}{2^2}+\frac{1}{2^3} = 0.875$	(1-0.07, 1-0.875) **(0.93, 0.125)**
8	(0.08, $\varphi_2(8)$)	1000	$\frac{0}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}+\frac{1}{2^4} = 0.0625$	(1-0.08, 1-0.0625) **(0.92, 0.9375)**
9	(0.09, $\varphi_2(9)$)	1001	$\frac{1}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}+\frac{1}{2^4} = 0.5625$	(1-0.09, 1-0.5625) **(0.91, 0.4375)**
10	(0.1, $\varphi_2(10)$)	1010	$\frac{0}{2^1}+\frac{1}{2^2}+\frac{0}{2^3}+\frac{1}{2^4} = 0.3125$	(1-0.1, 1-0.3125) **(0.90, 0.6875)**

better performance than LHS and crude Monte Carlo sampling techniques, and is at least 3–100 times faster for convergence (13).

A variant of the HSS sampling technique is the Latin hypercube Hammersley sampling (LHSS) (25). The aim of this sampling technique is to better utilize the 1D uniformity property of LHS and multidimensional uniformity property of HSS by coupling them. One dimensional uniformity analysis for Monte Carlo sampling, HSS, and LHSS is shown in Figure 14.

Other variants of HSS are Halton sequence sampling or shifted Hammersley, where the first variable is shifted and leaped Halton or Hammersley, where some of the cycles of these sequences are eliminated to improve efficiency for higher dimensional problems (24,26). As, the number of dimensions increase, the quasirandom sequences lose their uniformity properties. Therefore, to increase their performance, different quasirandom sequences could be combined and leaping procedure could be applied.

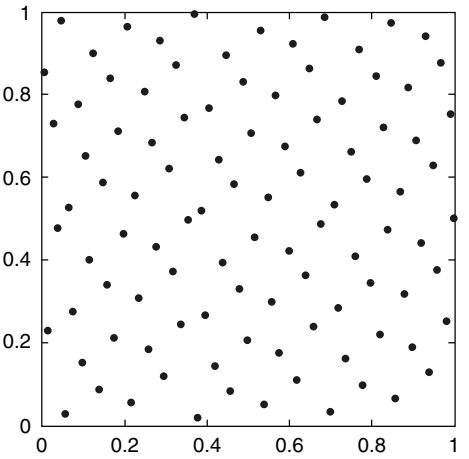

Fig. 11. Hammersley points on a unit square.

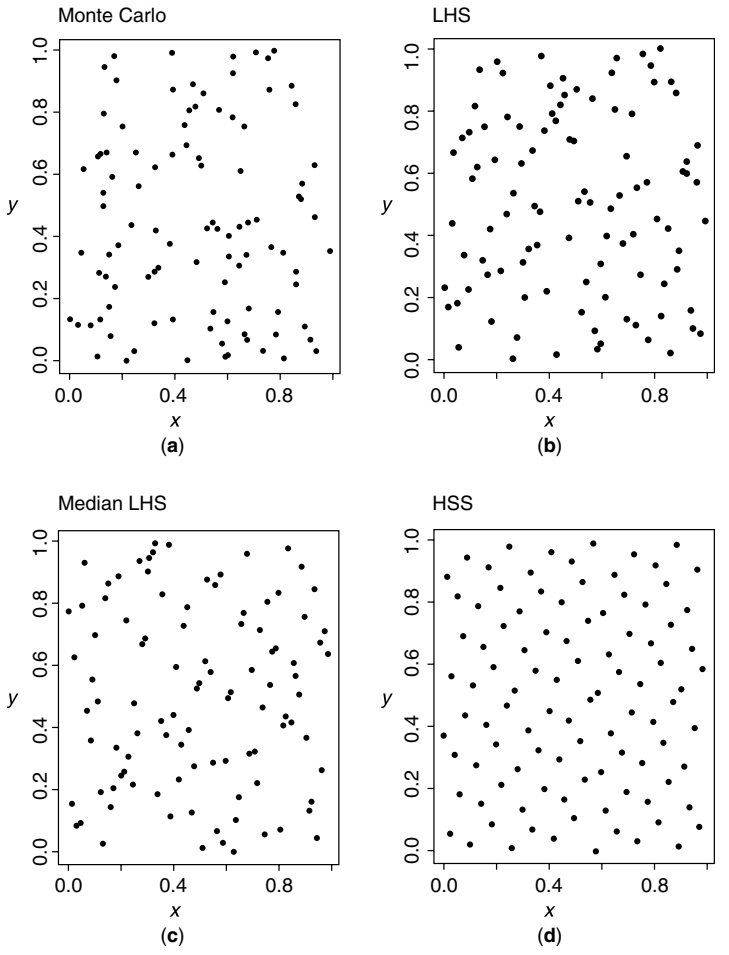

Fig. 12. 100 sample points on a unit square by (**a**) Monte Carlo sampling, (**b**) LHS, (**c**) MLHS, (**d**) HSS.

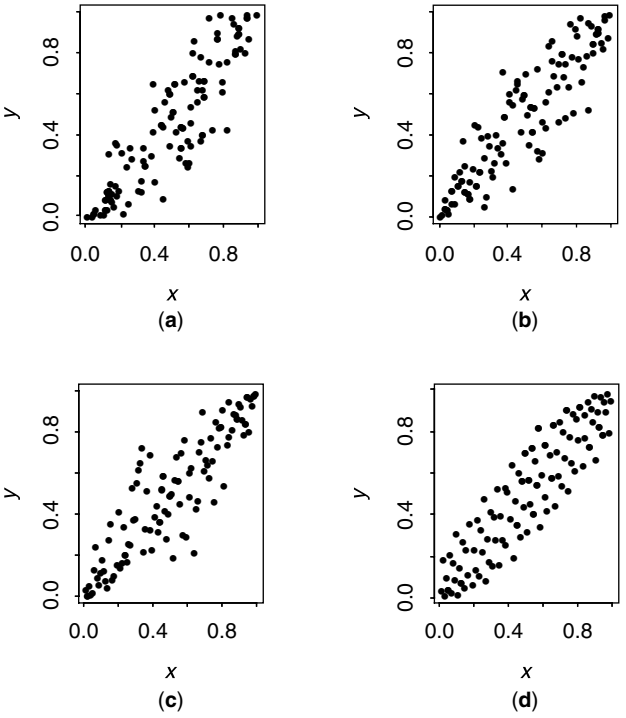

Fig. 13. Sample points (100) on a unit square with correlation 0.9 using (**a**) Monte Carlo sampling, (**b**) LHS, (**c**) MLHS, (**d**) HSS.

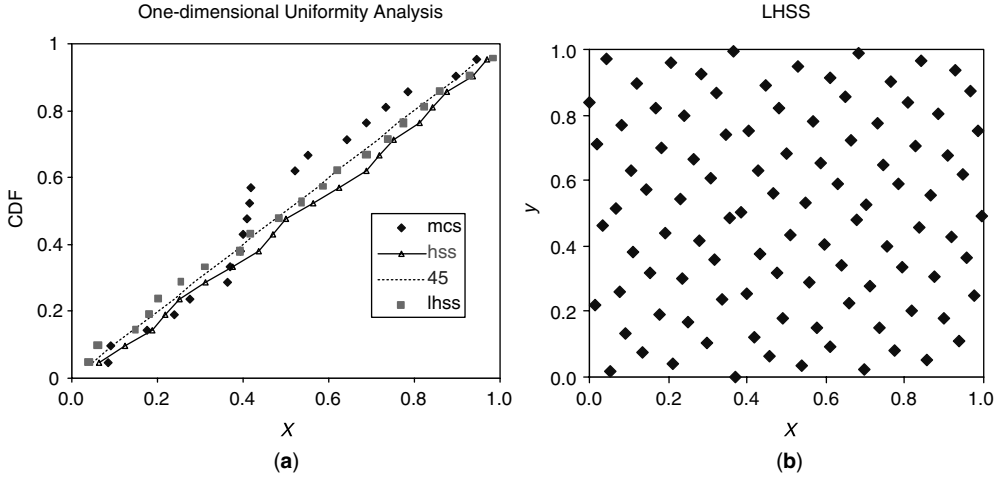

Fig. 14. (**a**) One-dimensional uniformity of sample points (20) on a unit square using MCS, HSS, and LHSS; (**b**) 100 sample points on a unit square generated by LHSS (25).

Parallelization of Monte Carlo and Quasi-Monte Carlo Methods. Monte Carlo and quasi-Monte Carlo sampling techniques discussed above are used to solve a variety of problems in computational chemistry as well as finance or economics, where complex models are used. Therefore, parallel computing to speed up calculations for these complex models is essential and these sampling techniques should be implemented in a parallel computer architecture, where independent simulations can be performed on different processors.

For reasons of efficiency, the random numbers generated for a parallel Monte Carlo simulation using different processors should be uncorrelated and should be generated independently. There are two basic parallelization techniques for generating random numbers. The first method assigns different random number generators for different processors. The second method assigns different substreams of one large random number generator to different processors. When the first method is used, it is possible that there are unknown correlations between the different random number generators in use. Alternatively, if the same random number generator with different parameters is used, one could also encounter similar problems. Hellekalek (27) addressed these issues related to random number generators for a parallel computer architecture with examples.

There are two variations of the second method. The first approach is a "leap-frog" technique, where a substream (x_{nL+j}) of lag L of the original sequence is assigned to the j-th processor, where $0<j<L-1$. The second approach is a "splitting" technique, where the original sequence is partitioned into L consecutive blocks. Each of the processors is assigned a different block and each block is defined by a unique seed. Both of these methods should be used with caution when the number of dimensions or the sample size is increased.

Mascagni provided a review of the parametrized versions of the pseudorandom number generators for parallel Monte Carlo applications, eg, linear congruential generators, linear matrix generators, shift-register generators, lagged Fibonacci generators, and inversive congruential generators (28).

Quasirandom sequences have also been used in parallel computing in recent years. In order to use quasirandom numbers in parallel, one can break up a single quasirandom number sequence into nonoverlapping blocks to be used in parallel processing elements. Comparison of parallel pseudorandom numbers and Sobol sequences has shown that the same kind of accuracy is achieved with the use of a quasirandom sequence in parallel. However, quasirandom sequence (where a block-based parallelization is used) converges to the same result in considerably less amount of time (29). Schmid and Uhl (30) also studied the parallelization of quasirandom sequences called the (t,s)-sequences. They have concluded that the block-based parallelization performs much better compared to leaping parallelization for numerical integration. In leaping parallelization, each processing element skips those points handled by other processing elements (leap-frogging).

2.3. Bayesian and Adaptive Methods. Bayesian probability theory was originally developed by Bayes (31). Bayesian and adaptive methods are used when the probability functions are not very accurate. The Bayesian method uses two steps. The first step is to identify the conceptual models and the distribution of model parameters. In the second step, the model results are compared

with existing observations through a structured probabilistic methodology. In the Bayesian context, a probability represents a degree-of-belief based on all the relevant information at hand. Classical statistical approaches are not very effective in predicting low frequency, rare, but consequential (eg, accidents or chemical spills) events. Bayesian theory could be applied to these cases (32). A Bayesian approach is also used for sensor fault detection (33).

One of the most important applications of Bayesian methods is to use Bayesian inference for model parameter estimation. This method uses the prior information about the parameters and the likelihood function to find the mode of the posterior distribution (34).

Bayesian Inference. Bayes theorem (31) states that the posterior probability distribution for an event is proportional to the prior distribution (knowledge) multiplied by the likelihood. If we denote D as the observed data and θ as the model parameters, we can write

$$P(D, \theta) = P(D|\theta)P(\theta) \tag{13}$$

In this equation, $P(D,\theta)$ is the joint probability distribution over all random quantities. This distribution is composed of two parts: a prior distribution $P(\theta)$ and a likelihood $P(D|\theta)$. In order to find the distribution of θ conditional on D, the Bayes theorem is used

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{\int P(\theta)P(D|\theta)d\theta} \tag{14}$$

This is called the posterior distribution of θ. The posterior expectation of a function $f(\theta)$ is

$$E[f(\theta)|D] = \frac{\int f(\theta)P(\theta)P(D|\theta)d\theta}{\int P(\theta)P(D|\theta)d\theta} \tag{15}$$

It is very difficult to integrate this expression and find $E[f(\theta)|D]$ especially in high dimensions, since for most applications the analytical solution is not available. One of the numerical approaches that have been used is the Markov chain Monte Carlo (MCMC) method described below.

Markov Chain Monte Carlo Method. Let X be a vector of k random variables with a distribution $\pi(.)$. In Bayesian inference, $\pi(.)$ will be a posterior distribution. Then, the task is to evaluate expressions of the form:

$$E[f(X)] = \frac{\int f(x)\pi(x)dx}{\int \pi(x)dx} \tag{16}$$

A Monte Carlo integration evaluates $E[f(X)]$ by drawing samples $\{X_t, t = 1, \ldots, n\}$ from $\pi(.)$ and approximates the integral by

$$E[f(X)] \approx \frac{1}{n}\sum_{t=1}^{n} f(X_t) \tag{17}$$

The sample averages are used to approximate the expectations. Markov chain Monte Carlo draws these samples by running a cleverly constructed Markov chain for a long time. A Markov chain can be defined as follows.

Suppose a sequence of random variables $\{X_0, X_1, X_2, \ldots, \}$ is generated such that at each time step, the next state is sampled from a distribution $P(X_{t+1}|X_t)$, which depends only on the current state of the chain X_t. This sequence is called a Markov chain and $P(X_{t+1}|X_t)$ is called the transition kernel of the chain. One of the transition kernels or updating schemes used in MCMC is the Gibbs transition kernel (35), which is a special case of the general framework of Metropolis and co-workers (36) and Hastings (37).

Many important implementation issues need to be considered for MCMC methods. These include the transition mechanism for the chain, the number of chains to be run, and their length and the choice of starting values. These issues are discussed in a review by Brooks (38). The MCMC methods and their implementation were also discussed in a textbook by Gilks and co-workers (39). There is also software called BUGS (Bayesian inference Using Gibbs Sampling) available at the World Wide Web (http://www.mrc-bsu.cam.ac.uk/bugs/) for analysis of complex statistical models using MCMC methods.

2.4. Other Sampling Techniques. Some other examples of non-Monte Carlo sampling techniques include, systematic sampling, cluster sampling, quota sampling, and multistage sampling. These sampling techniques are briefly described below.

Systematic Sampling. Systematic sampling is the selection of every nth element from a sampling frame. This sampling technique is also called the interval sampling, which means that there is a gap or interval between each selection. This technique is used in industry for quality control where a manufacturer might want to test an item from a production line at certain time intervals to make sure that it satisfies the product specifications, and the equipments and machines are working properly. A random starting point is selected and the sampling interval is chosen in a way that does not create a pattern that would threaten randomness. More information about systematic sampling can be found in Madow and Madow (40).

Cluster Sampling. In cluster sampling, the entire population is divided into clusters, or groups and a random sample is selected from these clusters. When the researcher does not have enough information about the individual members of a population, but can get a complete list of the groups or clusters, this sampling technique would be useful (41–43).

Quota Sampling. In quota sampling, the population is first divided into mutually exclusive subpopulations, just as in *stratified sampling*. Then the subjects are selected according to judgment or easy availability from each subpopulation. This sampling technique is often used in opinion polling and market research. This is not a random sample; therefore, statistical methods cannot be applied to measure the sampling error. A discussion on the validity of inferences made from quota sampling was presented by Smith (44).

Multistage Sampling. This sampling technique involves the selection of a sample in at least two stages. In the first stage large groups of clusters are selected and in the second stage population units are selected from the clusters

to derive a final sample. For an application of multistage sampling in epidemiology, refer to Refs. 45 and 46.

These sampling techniques are mostly designed for convenience and efficiency. However, they are not as accurate and many times the sampling error cannot be estimated by classical statistical techniques. They are more frequently used in areas such as market research, polling, or interviewing to infer some knowledge about a population. They are also encountered in epidemiology studies, where the causes and prevalence for certain diseases are statistically analyzed. However, they do not have a wide applicability in process systems engineering.

3. Uncertainty Analysis and Stochastic Modeling

The role of sampling in uncertainty analysis is indisputable and encompasses all application areas in process design, operation, and control. The uncertainties commonly encountered in chemical systems can be divided into two groups (47): (*1*) static uncertainties and (*2*) dynamic uncertainties.

3.1. Static Uncertainties. Static uncertainties are normally represented by probability distributions. Inclusion of uncertainties in a deterministic model results in a stochastic model. Stochastic modeling is an iterative procedure that consists of these four steps (48), as shown in Figure 15.

1. Uncertainty quantification that involve specifying uncertainties in key input parameters in terms of probability distributions.
2. Sampling distribution of the specified parameter in an iterative fashion.
3. Propagating the effects of uncertainties through the model.
4. Applying statistical techniques to analyze the results.

In the first step of stochastic modeling framework, uncertainties in key input variables are represented by probability distribution functions. An example of uncertainty characterization and quantification by probability distributions was presented by Kim and Diwekar (49) in a computer-aided molecular

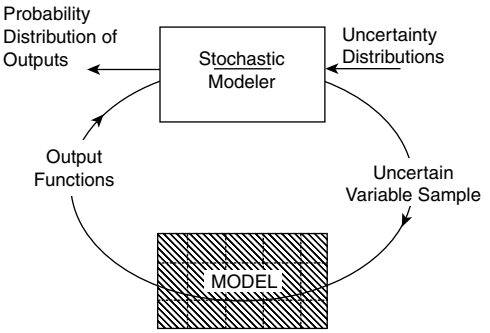

Fig. 15. Stochastic modeling framework.

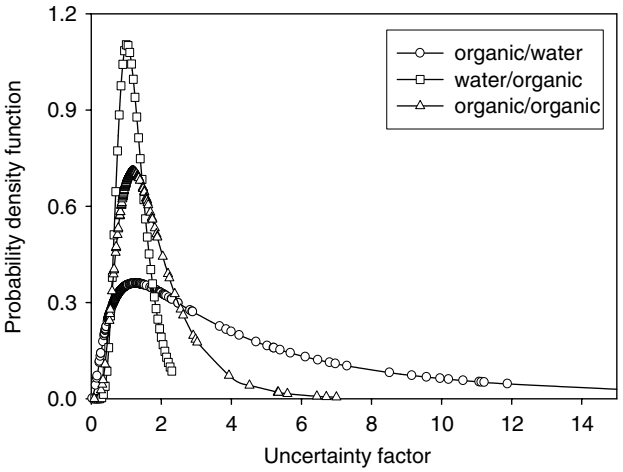

Fig. 16. Probability density functions for uncertainty factors for organic/water, water/organic, organic/organic families (50).

design (CAMD) problem. Discrepancies between the experimental data for predicting a thermodynamic property and the models are commonly encountered in CAMD. For example, Figure 16 shows the uncertainties in >1800 interaction parameters present in the UNIFAC activity coefficient model to predict solvent selection objectives for acetic acid separations. Uncertainty factors (UFs) were established as the ratio between the experimental and the calculated values of activity coefficients at infinite dilution γ^{∞} as defined in equation 18. Furthermore, uncertainty factors were divided into three categories based on the type of family: organic/water (lognormal distribution), water/organic (normal distribution) and organic/organic (lognormal distribution).

$$\mathrm{UF} = \frac{\gamma^{\infty}_{\mathrm{exp}}}{\gamma^{\infty}_{\mathrm{calc}}} \tag{18}$$

The type of distribution for an uncertain variable is a function of the amount of data available and the characteristic of the distribution function. The simplest distribution for an uncertain variable is a uniform distribution, which has a constant probability. This means that the uncertain variable can take any value within an interval $[a,b]$ with equal probability. On the other hand, if the uncertain variable is represented by a normal (Gaussian) distribution, there is a symmetric, but equal probability, that the value of the uncertain variable will be above or below a mean value. In log-normal or some triangular distributions, there is a higher probability that the value of an uncertain variable will be on one side of the median, resulting in a skewed shape. A beta distribution provides a wide range of shapes and is a very flexible means of representing variability over a fixed range. In some special cases, user-supplied distributions are used, eg, chance distribution. Different examples of probability distributions are given in Figure 17.

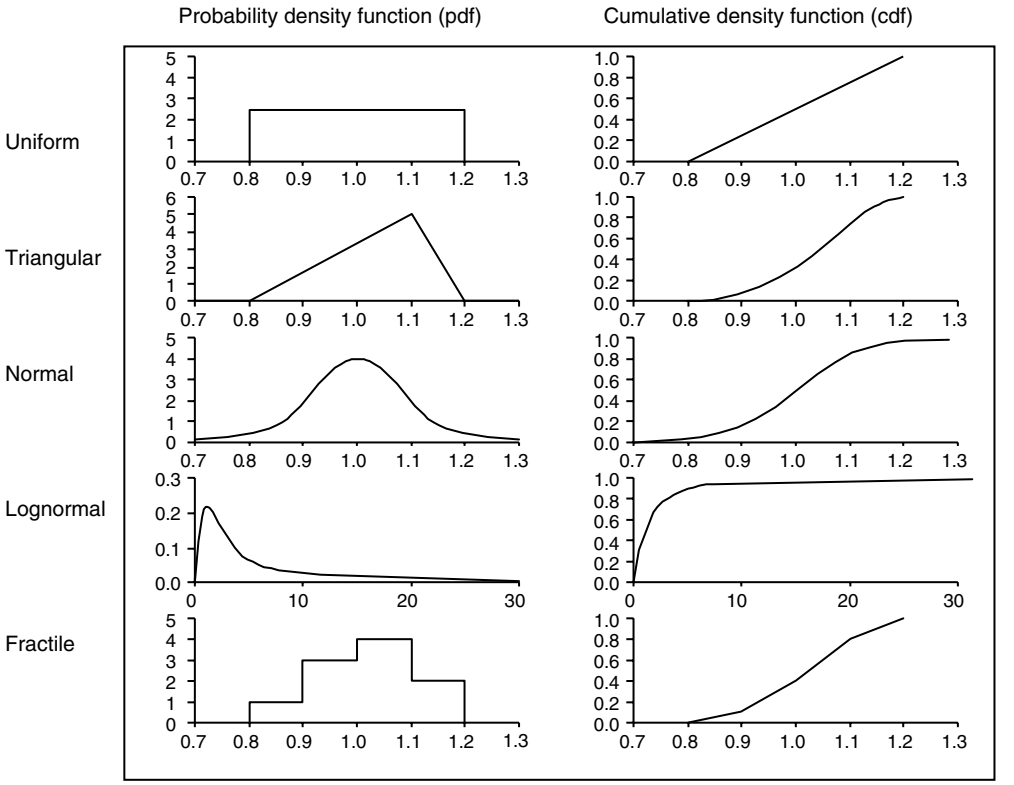

Fig. 17. Examples of probability distribution functions.

Once probability distributions are assigned to uncertain parameters, the next step is to perform a sampling operation from the uncertain parameter domain. Then, the uncertainties are propagated through the model. The stochastic modeler assigns the specified distributions to the input parameters and using sampling methods described in the previous section (eg, Monte Carlo, LHS, HSS), the sampled values of each uncertain variable are passed through the model. After a model simulation is run, the output variables of interest are collected. The simulation is then repeated for a new set of samples selected from the probabilistic input distributions. After all samples or observations have gone through the cycle for a specified number of times (typically 20–100 or more, depending on the accuracy sought by the user), the outputs are collected in terms of cumulative probability density functions.

3.2. Dynamic Uncertainties. Dynamic uncertainties are also ubiquitous in chemical systems, especially for batch processes. Due to the dynamic nature of these processes, even some of the static uncertainties are translated into dynamic uncertainties. An example of this is shown in Figure 18. This figure shows how the uncertainties in activity coefficients predicted by the UNIFAC method affect the time-dependent relative volatility profile in a batch distillation column (47). Despite this fact, a generalized method of treatment for dynamic

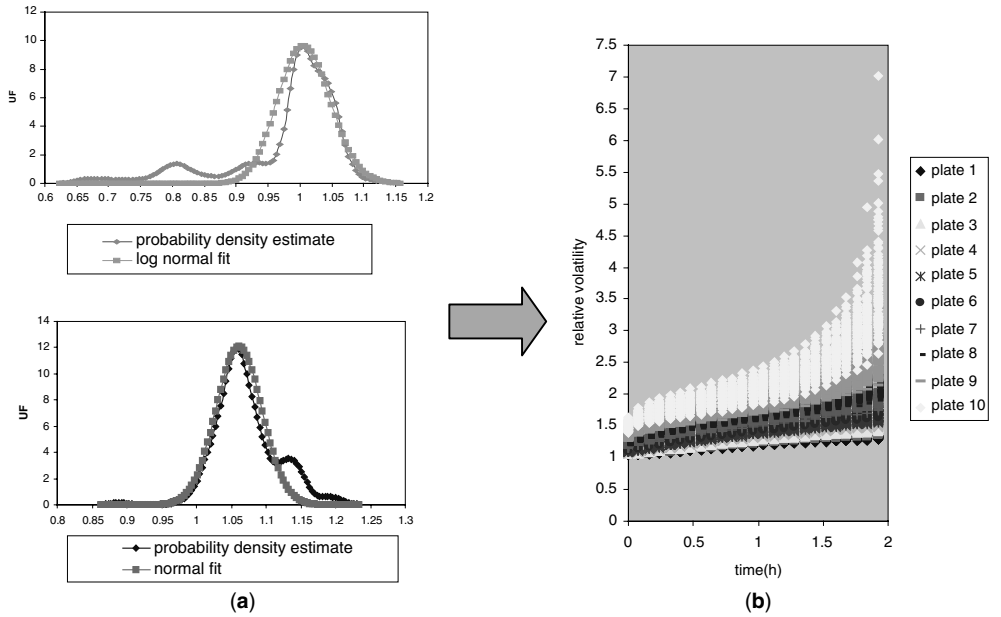

Fig. 18. (**a**) The uncertainties in activity coefficients predicted by UNIFAC. (**b**) The effect of static uncertainties on the relative volatility profile on each plate in a batch distillation column (47).

uncertainties for chemical systems was presented recently (47,51,52). This method is based on Ito processes and real options theory from finance literature.

Ito processes are a large class of continuous time stochastic processes. One of the simplest examples of a stochastic process is the random walk process. The Wiener process, also called a Brownian motion, is a continuous limit of the random walk and serves as a building block for Ito processes, through the use of proper transformations.

A Wiener process satisfies three important properties. First, it satisfies the Markov property. The probability distribution for all future values of the process depends only on its current value. Second, it has independent increments. The probability distribution for the change in the process over any time interval is independent of any other time interval (nonoverlapping), and third, changes in the process over any finite interval of time are normally distributed, with a variance that is linearly dependent on the length of time interval, dt. The general equation of an Ito process is given below:

$$dx = a(x,t)dt + b(x,t)dz \tag{19}$$

In equation 19, dz is the increment of a Wiener process, and $a(x,t)$ and $b(x,t)$ are known functions. There are different forms of $a(x,t)$ and $b(x,t)$ for various Ito processes. In this equation, dz can be expressed by $dz = \varepsilon_t\sqrt{dt}$, where ε_t is a random number drawn from a unit normal distribution.

The simplest generalization of equation 19 is the equation for Brownian motion with drift given by

$$dx = \alpha\, dt + \sigma\, dz \tag{20}$$

where α is called the drift parameter and σ is the variance parameter.

Other examples of Ito processes are the geometric Brownian motion with drift given in equation 21 and the geometric mean reverting process given in equation 22. Also, it has been shown that the relative volatility profile in Figure 18**b** can be represented by a geometric mean reverting process (47):

$$dx = \alpha x dt + \sigma x dz \tag{21}$$

$$dx = \eta(\bar{x} - x)dt + \sigma x dz \tag{22}$$

where η is the speed of reversion and $\bar{x}$ is the nominal level that x reverts to. In geometric Brownian motion, the percentage changes in x and $\Delta x/x$ are normally distributed (absolute changes are lognormally distributed). In geometric mean reverting processes, the variable may fluctuate randomly in the short run, but in the longer run it will be drawn back toward the marginal value of the variable. The expected change in x depends on the difference between x and $\bar{x}$. If x is greater (less) than $\bar{x}$, it is more likely to fall (rise) in the next short interval of time. The variance also grows with x.

The dynamic uncertainties in chemical processes (batch processes) could be represented by these Ito processes, depending on the character of uncertainty. For example, the thermodynamic uncertainties in batch processes were modeled using Ito processes and system nonidealities were easily distinguished (47). The parameters of the Ito process are estimated based on a regression analysis technique. For more details on Ito processes please refer to Refs. 52 and 53.

4. Efficiency Improvements in Optimization Algorithms

The role sampling plays in optimization algorithms extends beyond uncertainty analysis. Sampling accuracy is also crucial for deriving efficient algorithms for discrete optimization and multiobjective optimization problems, which will be described in this section.

4.1. Discrete Optimization. Discrete optimization problems involve discrete decisions and combinatorics. Discrete optimization problems are classified into groups, eg, integer programming (IP), mixed integer linear programming (MILP), and mixed integer nonlinear programming (MINLP). Many chemical engineering applications like chemical synthesis, process synthesis, planning, and scheduling involve discrete decision variables and mixed integer problems. Probabilistic combinatorial methods can be used to solve these problems. Examples of these methods are simulated annealing (SA) and genetic algorithms (GA). If the solution space is discontinuous or if the systems have large combinatorial explosion, these probabilistic methods provide an alternative to mathematical programming techniques, eg, branch and bound, generalized

Bender's decomposition (GBD) and outer approximations (OA) traditionally used to solve discrete optimization problems (52).

Simulated annealing is a heuristic combinatorial optimization method. Simulated annealing utilizes the analogy between the annealing procedure, where a metal cools and freezes into its minimum energy structure and the search for a minimal value in an optimization problem. In SA, the objective function (which is usually the cost) becomes the energy of the system. The goal is to minimize the cost (energy). Random permutations are generated to displace particles, which is analogous to moving the system to another configuration. If the configuration that results from the move has a lower energy state, the move is accepted. Otherwise the move is accepted according to the Metropolis criteria accepted with a probability $= \exp(-\Delta E/K_b T)$ (54). At high temperatures, a large percentage of uphill moves are accepted. As the temperature gets cooler, a small percentage of uphill moves are accepted. After the system has evolved to thermal equilibrium at a given temperature, the temperature is lowered and the annealing process continues until the system reaches the "freezing" temperature. Painton and Diwekar (55) used the SA technique to improve the performance of space nuclear power plants.

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. The G_{ij} of the conventional SA algorithms rely on pseudorandom number generators, eg, Monte Carlo sampling, which result in clustered moves over the configurational space. Therefore, a larger number of moves or generations are needed to cover the configurational space more evenly, which results in a longer Markov chain length (ie, number of moves) at each temperature level. As mentioned earlier, HSS technique can generate quasirandom samples showing k-dimensional uniformity properties. The HSS technique was used to develop a new SA algorithm called the efficient simulated annealing (ESA). Since HSS generates more uniform samples in multivariate space, it requires fewer numbers of moves to approximate ideal probabilities.

Figure 19 shows the trajectories of the objective value for the test function $f(y) = \sum_{i=1}^{10} y_i^2$ with different Markov chain lengths. The ESA found the solution with a Markov chain length of 45 at each temperature while the traditional SA needed a Markov chain length of 75 to reach the same solution. Here ESA was found to be $\sim$30–54% more efficient than conventional SA (56).

Genetic algorithms (GA) are also used for combinatorial optimization problems. The GAs follow a search procedure based on Darwin's theory of evolution and the idea of survival of the fittest. The GAs begin with a set of solutions (represented by chromosomes) that is called the initial population. The population for the next generation is selected according to a randomized selection procedure involving four operators: (*1*) reproduction; (*2*) crossover; (*3*) mutation and immigration, and the fittest individuals are selected from the population for the next generation. This procedure is repeated until a stopping criterion, such as number of populations or improvement of the best solution, is satisfied. Conventional GAs also use Monte Carlo sampling based on pseudorandom numbers for generating the initial population and various genetic operators. Similar to the ESA, an

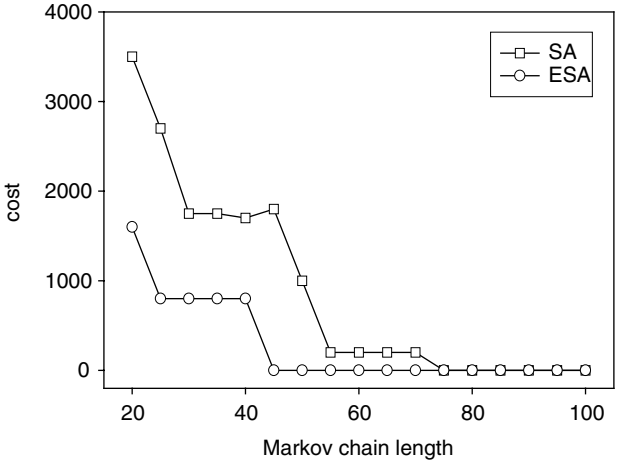

Fig. 19. The comparison between SA and ESA (56).

efficient genetic algorithm (EGA) based on the use of Hammersley sequence sampling technique was developed by Diwekar and Xu (57), in order to improve the efficiency of conventional GAs. Table 6 shows the comparison of GAs for problems varying in complexity and size.

4.2. Optimization Under Uncertainty. Optimization under uncertainty refers to the branch of optimization problems where there are uncertainties involved in the data or the model, and is popularly known as stochastic programming or stochastic optimization problems. The generalized stochastic framework to solve optimization problems under uncertainty involves two recursive loops: (*1*) the sampling and (*2*) the optimization loop. A schematic representation of this stochastic framework is shown in Figure 20. By interchanging the position of the sampling loop, two kinds of solution procedures could be obtained. These are called "here and now" and "wait and see" problems. "Here and now" problems yield optimal solutions to achieve a given level of confidence. On the other hand, "wait and see" problems involve a category of formulations that show the effect of uncertainty on optimum design. A here and now problem replaces a deterministic model by an iterative stochastic model with sampling loop representing the discretized uncertainty space as shown in Figure 20. A wait and see problem involves deterministic optimal decision at each scenario

Table 6. **Efficiency Improvement in GA Using the HSS Technique**[a]

Problems	Number of dimensions (nd)	Optimal value	Generation		Efficiency improvement, %
			MGA (Monte Carlo)	EGA (HSS)	
problem 1	10	0	15	4	73.33
	20	0	43	10	76.74
problem 2	3–11	0	9	6	33.33
problem 3	5	−1	176	83	52.84

[a]See Ref. 57.

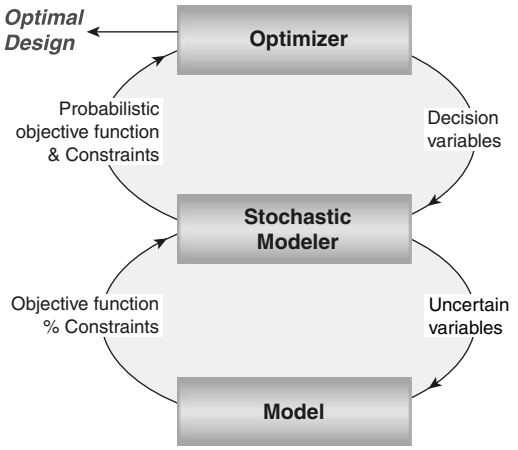

Fig. 20. Stochastic programming framework.

or random sample, equivalent to solving several deterministic optimization problems. A simple example for a stochastic program is given below in Example 6.

Example 6: News Vendor (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 that is uncertain. The selling price is s_p cents per paper. For a specific case, the weekly demand is shown in Table 7.

For this case, the cost of each paper is $c = 20\,¢$ and selling price s_p is 25 ¢. Solve the problem if the news vendor knows the demand uncertainties given above in Table 7, but does not know the demand curve for the following week *a priori* (Table 8). Assume no salvage value $s = 0$, so that any papers bought in excess of demand are simply discarded with no return.

Solution: In this example problem, the aim is to find the number of papers the vendor must buy (x) to maximize the profit. Let r be the effective sales and w be the excess that is going to be thrown away. This is a stochastic programming problem where action (x) is followed by observation (profit) and reaction (or recourse) (r and w). It is known that any papers bought in excess are just thrown away. Therefore, one should minimize the excess, but increase the sells. Our first instinct to solve this problem is to find the average demand and find the optimal supply x corresponding to this demand. Since the average demand from Table 7 is 70 papers, $x = 70$ should be the solution. However, with this solution where supply is 70 papers/day, the newsvendor will be making a loss of 50 ¢/week.

Table 7. **Weekly Demand Uncertainties**

j	Demand, d_j	Probability, p_j
1	50	5/7
2	100	1/7
3	140	1/7

Table 8. **Weekly Demand**

i	Day	Demand, $(u)\, d_i$
1	Monday	50
2	Tuesday	50
3	Wednesday	50
4	Thursday	50
5	Friday	50
6	Saturday	100
7	Sunday	140

This is probably 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. The above information can be transformed for daily profit as follows:

$$\text{Profit} = -cx + 5/7 s_p d_1 + 1/7 s_p(x) + 1/7 s_p(x) \tag{23}$$
$$if\ d_1 \le x \le d_2$$
or
$$\text{Profit} = -cx + 5/7 s_p d_1 + 1/7 s_p(d_2) + 1/7 s_p(x) \tag{24}$$
$$if\ d_2 \le x \le d_3$$

Note that the problem represents two equations for the objective function, equations 23 and 24, making the objective function a discontinuous function and is no longer an LP (linear program). The optimal solution to this problem is $x = d1 = 50$ providing the news vendor with an optimum profit of 1750 ¢/week.

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 demand, ie, $x = 70$ as the solution, we obtain a loss of 50 ¢/week. Therefore, the VSS is $1750 - (-50) = 1800$ ¢/week.

Now consider the case, where the vendor knows the exact demand (Table 8) *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.

$$\underset{x_i}{\text{Maximize}}\ \text{Profit}_i = -cx_i + \text{Sales}(r, w, d) \tag{25}$$
$$\text{Sales}\,(r, w, d_i) = s_p r_i + s w_i$$
$$r_i = \min(x_i, d_i) \tag{26}$$
$$= x_i, if\ x_i \le d_i \tag{27}$$
$$= d_i, if\ x_i \ge d_i \tag{28}$$
$$w_i = \max(x_i - d_i, 0)$$
$$= 0, if\ x_i \le d_i \tag{29}$$
$$= x_i - d_i, if\ x_i \ge d_i \tag{30}$$

Table 9. **Supply and Profit**

i	Day	Supply, x_i	Profit, ¢
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

Here we need to solve each problem (for each i) separately, leading to the following decisions shown in Table 9.

One can see that the difference between the two values, (*1*) when the news vendor has the perfect information (\$2450 ¢/week) and (*2*) when he does not have the perfect information (\$1750 ¢/week), but can represent it using probabilistic functions, is the Expected Value of Perfect Information (EVPI). The EVPI is 700 ¢/week for this problem.

Both here and now and wait and see problems require representation of uncertainties in the probabilistic space, and then propagation of these uncertainties through the model to obtain probabilistic representation of output. Here, sample approximation methods and sampling accuracy are often used to derive new algorithms for optimization under uncertainty.

Sample Approximation Methods. As stated earlier, the stochastic programming formulations often include some approximations of the underlying probability distribution. The disadvantage of sampling approaches that solve the γth approximation completely is that some effort might be wasted on optimizing when approximation is not accurate (58). For stochastic linear programming the L-shaped method is a commonly used technique (59). 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 and Infanger (60), 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 (61). For more details on stochastic programming please refer to Refs. 52 and 58.

Sample average approximation methods have also been used to reduce the computational time and increase accuracy for stochastic process design problems. Wei and Realff (62) presented a method that involves two algorithms: optimality gap method (OGM) and confidence level method (CLM), to solve convex stochastic mixed-integer nonlinear problems. A smaller sample size is used to make decisions (with several replications) and a larger one is used to reevaluate the objective value with the decision variables fixed. The sample sizes and replication number are increased until a stopping criterion is satisfied. In the OGM algorithm, the sample sizes are increased until the optimality gap of each upper

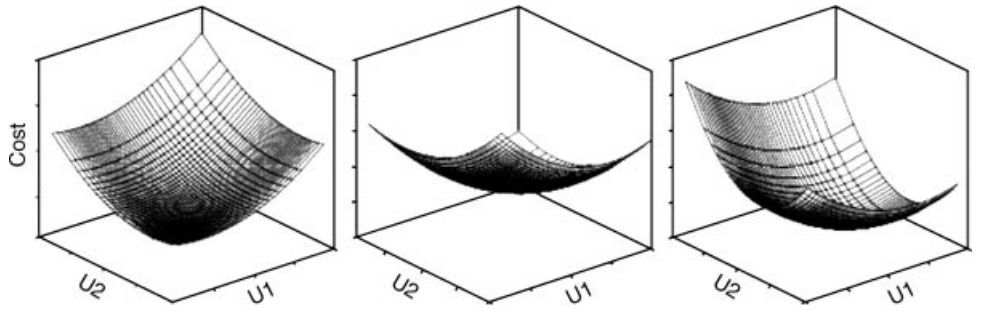

Fig. 21. Uncertainty space at different optimization iterations (52).

and lower bound is sufficiently small; in the CLM algorithm, the sample sizes are increased until an overall accuracy probability is within a certain tolerance.

Sampling Accuracy and Optimization. In almost all stochastic optimization problems, the major bottleneck is the computational time involved in generating and evaluating probabilistic functions that represent the objective function and constraints. The number of samples required for a given accuracy in stochastic optimization problem depends on certain factors, eg, type of uncertainty and the point values of the decision variables (55). Especially, for optimization problems, the number of samples required depends on the location of the trial point solution in optimization space. Figure 21 shows how the shape of the surface over a range of uncertain parameter values changes at different optimization iterations. Therefore, the selection of the number of samples is an important step which ultimately decides the accuracy of the optimal solution in stochastic programming.

For the solution of stochastic integer programming problems, variants of SA and GA have been developed.

Stochastic Annealing and Stochastic Genetic Algorithms. The stochastic annealing (STA) algorithm (53,63,64) is a variant of simulated annealing and is used to optimize stochastic integer programming problems. The STA provides an improvement over SA by obtaining both the decision variables and the number of samples required for the optimization problem. For balancing computational efficiency and solution accuracy, a penalty function is introduced in the objective function to ensure that the algorithm selects greater number of samples as the solution nears optimum value.

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

$$\min Z(\cos t) = P(x;u) + b(t)\varepsilon \tag{31}$$

The weighting function $b(t)$ can be expressed in terms of the temperature levels (t) and is given by $b(t) = b_0/k^t$ where b_0 and k are constants. At high

temperatures, the sample size is small, and the algorithm is mainly exploring the functional topology to identify regions of optima. As the system gets cooler, the algorithm searches for the global minimum and more accurate estimates of the objectives/costs are needed and this requires more samples. The error bandwidth of the Monte Carlo samples ($\varepsilon_{\mathrm{MCS}}$) is estimated by the central limit theorem.

Using the HSS technique for the generation probability G_{ij}, Kim and Diwekar (56) developed the ESTA. This algorithm uses the central limit theorem to evaluate the sampling errors and uses the HSS technique for generation probability. Another variant of STA is the Hammersley stochastic annealing algorithm (HSTA). This algorithm was presented by Kim and Diwekar (56) and it uses (*1*) HSS for the generation probability G_{ij} in annealing procedure; (*2*) HSS for the inner-sampling loop, where number of samples N_{samp} are determined; and (*3*) the HSS specific error bandwidth ($\varepsilon_{\mathrm{HSS}}$). The error bandwidth for the Hammersley sequence samples ($\varepsilon_{\mathrm{HSS}}$) is given by a fractal dimension analysis (64,65). This methodology uses the k-dimensional uniformity properties of HSS technique and the HSS error bandwidth to achieve a trade-off between accuracy and efficiency. Example 7 describes the steps of the HSTA algorithm in a numerical example.

Example 7: In order to show the applicability of the HSTA algorithm, the following test function is used:

$$\begin{aligned} \min & \sum_{i=1}^{2}(u_i \times y_i^2) \\ & -20 \le y \le 20 \\ & u \sim N(0.5, 0.16) \end{aligned} \tag{32}$$

The initial configuration of y is (20,20) and the uncertainty variable u follows a normal distribution with mean 0.5 and a standard deviation of 0.16. The simulation conditions are

$$\begin{aligned} &\text{Initial temperature } (T_{\mathrm{initial}} = 1) \\ &\text{Temperature decrement } (\alpha = 0.85) \\ &\text{Markov chain length } (I = 20) \\ &\text{Initial number of samples } (N_{\mathrm{samp}} = 30) \\ &b_0 = 0.005;\ k = 0.940 \end{aligned}$$

Step 1: Generate 20 (Markov chain length) sets of random numbers using HSS. Two random numbers are used for generation probability G_{ij}, three random numbers (H_k) are used for determining N_{samp} and one random number A_{ij} is used for the Metropolis criteria.

Step 2: Generate the next configuration. If $G_{ij,1}$ for random selection is <0.5, then y_1 is selected for random bump. If $G_{ij,2}$ is <0.5, then the value of

selected y_i is decreased. If the bumped values reside outside the bounds, the random bump is increased to the original y_i value. For example, for this case the G_{ij} are 0.2031 and 0.6914. Therefore, y_1 is increased to 21. But this is outside the bounds and thus y_1 becomes 19.

Step 3: Determine the number of samples N_{samp} for trade-off between solution accuracy and efficiency. To determine N_{samp}, three random numbers (H_k) generated by the HSS technique is used. If H_1 is <0.5, then $N_{\text{samp}} + 5 \times H_2$ becomes new N_{samp}. Otherwise, $N_{\text{samp}} - 5 \times H_2$ becomes a new one. The new N_{samp} for this case is 28.

Step 4: Generate N_{samp} samples for the uncertain variable u. Evaluate the probabilistic objective constraints, expected value and penalty function. The expected value is 219.36 and the penalty term is $1.3E$-5.

Step 5: Determine is the current configuration is accepted or rejected based on Metropolis criterion. If $\Delta E[z] \leq 0$ then the current configuration is accepted. If $\Delta E[z] > 0$ then the move is accepted with a probability $\exp[-\Delta/T]$. Since $\Delta E[z] = \Delta E[z]_{\text{new}} - \Delta E[z]_{\text{old}}$ is negative, the current configuration is accepted.

Step 6: Repeat the steps from 2 to 5 if the current iteration point is smaller than Markov chain length. In Table 10, 20 iterations at the first temperature level (TL) are shown.

Step 7: Check the stopping criteria and decrease temperature. If any of the stopping criteria are satisfied, then the simulation is terminated with a successful result. Otherwise, new temperature becomes $T = \alpha T$, and simulation goes back to Step 2. Table 11 shows the simulation results with respect to TL, where the optimum solution is reached at the 10th temperature level.

The HSTA algorithm is a useful tool for solving large scale combinatorial optimization problems under uncertainty and was applied to computer aided molecular design problems (56).

Table 10. **Configurations at the First TL**

I	y_1	y_2	$G_{i,j,1}$	$G_{i,j,2}$	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

Table 11. **HSTA Simulation Results**

Temperature level (TL)	$E[z]$	$b(t)\varepsilon$	y_1	y_2
1	121.28	2E-5	14	15
2	53.41	4E-5	9	10
3	12.04	4E-5	4	5
4	0.00	3E-5	0	0
5	0.59	3E-5	1	−1
6	0.59	3E-5	1	−1
7	0.31	3E-5	1	0
8	0.31	3E-5	1	0
9	0.31	4E-5	1	0
10	0.00	5E-5	0	0

A similar approach was also applied to genetic algorithms by Diwekar and Xu (57). First the stochastic genetic algorithm (SGA) was developed that employs Monte Carlo sampling for stochastic optimization problems. Then efficient stochastic genetic algorithm was developed that uses HSS technique and Monte Carlo confidence intervals. Finally, the Hammersley stochastic genetic algorithm (HSGA) was introduced that uses HSS technique and HSS specific error bandwidth to achieve a trade-off between accuracy and efficiency. The HSGA displayed the best performance among these algorithms. The performance of these three algorithms is compared in Figure 22 for the test function:

$$f(x,y,\xi) = \sum_{i}^{ND}\left(\xi_i x_i - \frac{i}{ND}\right)^2 + \sum_{i}^{ND}\xi_i y_i^2 - \prod_{i}^{ND}\cos(4\pi\xi_i y_i) \tag{33}$$

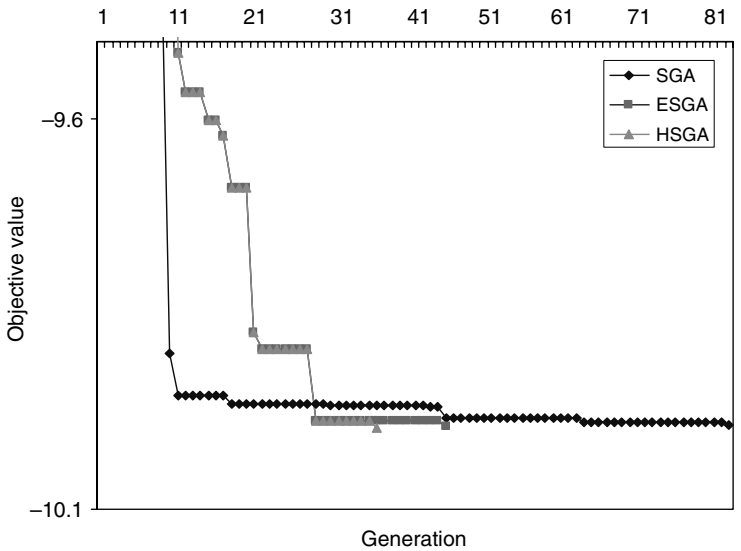

Fig. 22. Comparison of performance and convergence path for SGA, ESGA, and HSGA.

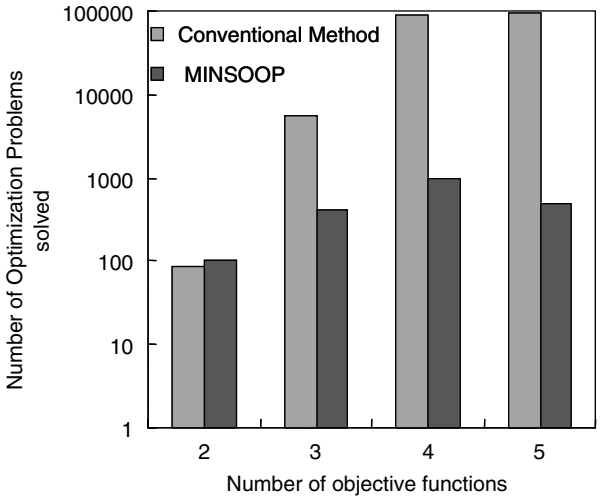

Fig. 23. Computational speed-up through MINSOOP (67).

4.3. Multiobjective Optimization. Multiobjective optimization (MOP) problems deal with conflicting and different objectives. They are commonly used where engineers are not only looking for low cost options but also trying to reduce the environmental and health impacts and risk and improve the reliability and safety of the plant. A generalized multiobjective optimization problem can be formulated as follows:

$$\begin{aligned} \min \quad & Z = Z_i, i = 1, \ldots, p \quad p \geq 2 \\ \text{s.t.} \quad & h(x,y) = 0 \\ & g(x,y) \leq 0 \end{aligned} \tag{34}$$

where x and y are continuous and discrete decision variables, and p is the number of objective functions. The functions $h(x,y)$ and $g(x,y)$ represent equality and inequality constraints, respectively. There is a large array of analytical techniques to solve this MOP problem; however, the MOP methods are generally divided into two basic types: preference-based and generating methods. Preference-based methods such as goal programming attempt to quantify the decision-maker's preference, and with this information, the solution that best satisfies the decision-maker's preference is then identified (2,52). As is well known, mathematics cannot isolate a unique optimum when there are multiple competing objectives. Mathematics can at most aid designers to eliminate design alternatives dominated by others, leaving a number of alternatives in what is called the Pareto set (66). Generating methods, eg, the weighting method and the constraint method, have been developed to find the exact Pareto set or an approximation of it. For each of these designs, it is impossible to improve one objective without sacrificing the value of another relative to some other design alternative in the set. From among the dominating solutions, it is then the design that is the most appropriate for that particular purpose is selected. At issue is an effective means to find the members of

the Pareto set for a design problem, especially when there are more than two or three objectives; the analysis per design requires significant computations to complete, and there are an almost uncountable number of design alternatives. A pure algorithmic approach to solving is to select one to minimize while the remaining objectives are turned into an inequality constraint with a parametric righthand side, L_k. The problem takes on the following form:

$$\begin{aligned} &\min \ Z = Z_j \\ \text{s.t.} \ &h(x,y) = 0 \\ &g(x,y) \le 0 \\ &Z_k \le L_k, k = 1, \ldots, j-1, j+1, \ldots, p \end{aligned} \tag{35}$$

where Z_j is the chosen jth objective that is to be optimized. Solving repeatedly for different values of L_k chosen between the upper, $Z_U(j)$ and the lower, $Z_L(j)$, bounds leads to the pareto set. This is the basis of the MINSOOP algorithm.

MINSOOP (Minimizing Number of Single Objective Optimization Problems) algorithm was developed by Fu and Diwekar (67) to address multiobjective optimization problems based on HSS technique. The steps for a multiobjective optimization problem with k objectives (to be minimized) are listed as follows:

Step 1: Solve k single objective optimization problems individually with the original constraints of a multiobjective problem to find the optimal solution for the individual k objectives.

Step 2: Compute the value of each of the k objectives at each of the k individual optimal solutions. In this way, an approximation of the potential range of values for each of the k objectives is determined and saved in a table (called payoff table). The minimum possible value is the individual optimal (minimizing) solution. The approximate maximum possible value of the Pareto set is the maximum value for that objective found when minimizing the other $k - 1$ objectives individually.

Step 3: Select a single objective (Z_l) to be minimized. Transform the remaining $k-1$ objectives into equality constraints of the form $Z_i \le {}_i$, $i = 1, \ldots, k$, $i \ne l$ and add these new $k - 1$ constraints to the original set of constraints. Then the original multiobjective optimization problem is transformed into a family of single objective optimization problems with parametric right hand sides.

Step 4: Select a desired number of single objective optimization problems to be solved to represent the Pareto set. Using the HSS technique to generate the desired number of combinations of the inequality constraint values $\varepsilon_i, \ldots, \varepsilon_{l-1}, \varepsilon_{l+1}, \ldots, \varepsilon_k$ within the range determined in step 2.

Step 5: Solve the constrained problems set up in step 4 for every combination of the right hand side values determined in step 3. These feasible solutions form an approximation for the Pareto set.

The uniformity property of HSS technique is crucial for the success of MINSOOP algorithm. Figure 23 shows how the MINSOOP algorithm improves efficiency for a simple nonlinear convex optimization problem as the number of objectives increases. Similar improvements are noted in multiobjective genetic algorithms based on the uniformity property of HSS (68).

5. Applications of Sampling Techniques in Chemical Systems

The life cycle of a chemical manufacturing process extends from product discovery, raw material selection (chemical synthesis) and process development (process synthesis and design) to process operation, planning and management that involve tasks, such as scheduling, supply chain, and process control. Contemporary process design approaches require engineers to not only look for low cost options, but also include several other criteria, eg, reliability, flexibility, operability, controllability, environmental and ecological impacts, safety and quality into different stages of analysis and design. This results in additional complexities and uncertainties. This section will present some of the applications of sampling techniques in product discovery, chemical synthesis, process synthesis and process operations ranging from scheduling, supply chain and process control. The role of sampling techniques in risk and reliability analysis will also be discussed.

5.1. Product Discovery and Design: Computational Chemistry and Molecular Simulations. At the discovery stage of a chemical process, computational chemistry, molecular modeling, and simulations are widely used in chemical and pharmaceutical industries. For the design and discovery of new molecules or drug compounds, Monte Carlo or molecular dynamics simulations are applied to estimate the physical, chemical, biological, and toxicological properties of interest. A review of molecular modeling and simulation techniques can be found in Refs. 15 and 69. These methods depend heavily on random number generators and sampling. The following discusses the role of sampling in molecular simulations where Monte Carlo methods are predominant.

In Monte Carlo molecular simulations, particles are randomly selected and moved by a random extent and the energy change of the system is analyzed. These random perturbations on the system configuration are accepted according to Metropolis criterion, with a probability proportional to the Boltzmann constant. This forms the basis of Metropolis Monte Carlo (MMC) method that is the first approach in increasing efficiency of Monte Carlo molecular simulations by using the "importance sampling" concept. In importance sampling, a biased distribution is used to obtain more samples from a region of importance. Boltzmann distribution function is such a distribution used in MMC where system configuration states that make substantial contributions to the ensemble averages are generated. The MMC method requires large number of samples to generate accurate property estimations and is computationally intensive especially for large number of molecules and complex fluids. Therefore, many researchers have worked on sampling techniques in order to speed up the calculations and cover the configurational space more efficiently. Some examples of these biased sampling techniques are configurational-bias Gibbs ensemble (70)

and its variants (71–73), and non-Boltzmann biasing techniques (74,75). Despite the fact that, the literature is abundant in examples of algorithms for efficiency improvement for Monte Carlo simulations, most of these methods are derived based on the importance sampling (biased sampling) principle and are often problem specific and offer customized solutions for particular systems.

Recently, a universal approach for increasing efficiency in molecular simulations using the HSS technique was presented (76). Pseudorandom numbers are used in MMC method for performing the random moves for the molecules and for acceptance probability. This new technique replaces the pseudorandom numbers in a systematic way by quasirandom samples of HSS, to speed-up the simulations and to increase the accuracy. While replacing the pseudorandom numbers with quasirandom samples, the k-dimensional uniformity property of HSS technique was maintained and exploited to cover the configurational space more efficiently. This method was used to estimate thermodynamic and biological/toxicological properties of chemicals, more specifically octanol–water partition coefficients. This new framework provided a better way to predict octanol–water partition coefficients in terms of prediction accuracy and computational efficiency (number of cycles), as shown in Figure 24. Also note that this proposed approach can be used in conjunction with the biased MCS (importance sampling) strategies presented in the literature and is not restricted to specific applications.

5.2. Chemical Synthesis. Process design starts with chemical synthesis in a laboratory where a chemical pathway from reactants to products is defined. This involves the search for molecules possessing desired physical, chemical, biological properties. Computer aided chemical synthesis relies on the group contribution methods (77,78) which assign numerical values to functional groups forming each molecule, through experimental data and theoretical methods. It is possible to calculate a wide range of characteristics for any given chemical by combining these functional groups.

A basic diagram of computer aided molecular design (CAMD) is shown in Figure 25. The starting point in CAMD is a set of functional groups. All possible combinations of these functional groups are explored to generate molecules that

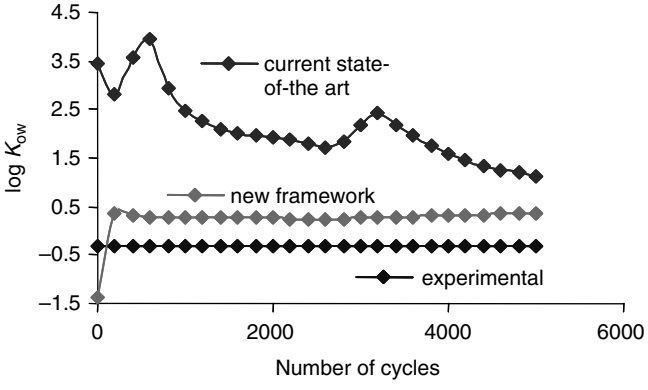

Fig. 24. Octanol–water partition coefficient for propanol predicted by molecular simulations.

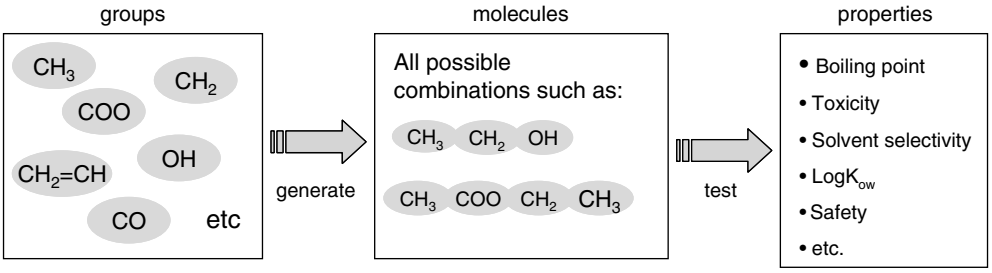

Fig. 25. A basic diagram of computer-aided molecular design.

satisfy feasibility constraints. The properties of each group and the interaction parameters between groups can be obtained theoretically, experimentally, or by applying statistical regression techniques. Once molecules are generated, the properties of these molecules are inferred from the properties of the functional groups structuring them. If the generated molecule satisfies certain criteria, then it is added to the list of candidate molecules. This method can generate a list of candidate molecules for any purpose with reasonable accuracy within a moderate time scale.

There are three main CAMD approaches: generation-and-test, mathematical optimization and combinatorial optimization approaches (79). All methodologies for CAMD are subject to uncertainties due to experimental errors, imperfect theoretical models/parameters, and inadequate knowledge of the systems. Furthermore, group parameters may not be available and current group contribution models (GCM) cannot estimate all necessary properties.

Uncertainties in CAMD have been addressed in various publications. For example, Maranas (80) studied polymer design with optimal thermophysical and mechanical properties. These properties are estimated based on group contribution methods and there are always discrepancies between the experimental data and the data predicted by group contribution method. In order to model the uncertainties in group contribution parameters, probability distribution functions are utilized, which results in a chance constrained formulation. These chance constraints represent the probability of meeting the target values of properties. The solution to the optimal molecular design problem under uncertainty has at least an α chance of meeting performance objectives and β chance of maintaining property values within their designated bounds. Since the formulations provided by Maranas (80) involve probability terms, this poses a problem of integrating multivariate probability density distributions. In order to overcome the computational burden, stochastic constraints are transformed into equivalent deterministic ones. This allows reaching an exact solution to the resulting convex MINLP formulation.

Tayal and Diwekar (81) also addressed property prediction uncertainty in polymer design and presented a generalized stochastic framework based on HSS and stochastic annealing to solve this problem. Because of its increased computational efficiency, this framework is applicable to nonlinear or even black box property prediction models, nonlinear objective function and constraints and stable and nonstable distributions for the uncertain variables. It

also provides a set of solutions instead of only one solution, which gives flexibility to the designer.

Kim and Diwekar (49,50,82) applied the CAMD approach to the selection of environmentally benign solvents under uncertainty for extraction. Uncertainties in property prediction models were quantified using available experimental data as shown earlier in Figure 16. The HSTA algorithm was implemented (56) to solve this combinatorial optimization problem. This algorithm makes use of the efficient Hammersley sequence sampling technique for updating discrete combinations, reducing Markov chain length and for determining the number of samples automatically.

The problem of environmentally benign solvent selection was also studied using genetic algorithms (57,83). Hammersley stochastic genetic algorithm was developed, which outperforms the HSTA algorithm by choosing solvents with better targeted properties in less computational time.

5.3. Experimental Design, Model Building, and Parameter Estimation. Experimental design is used frequently by researchers for assessing the performance of a new catalyst, determining a reaction mechanism, or determining the best operating conditions for a chemical plant. Experimental design also affects the fidelity of the fundamental and semiempirical models developed and model parameters estimated using experimental data for the process at hand. Reliable models offer competitive advantage to industries for model-based process design, operations, and control.

In experimental design, it is desired that the experimental region that is sampled generates the maximum amount of information for determining the correct model from a set of candidates and estimating the parameters of this model with the greatest precision.

When building models one uses some prior information, such as physical, chemical, or biological laws, and propose possible model candidates for the process under consideration. These models may have parameters that have a physical meaning, eg, the kinetic constants for a chemical reaction that one wishes to calculate with maximum precision. Usually, these models consist of mixed differential and algebraic equation systems.

Model building consists of the following three stages (84):

Stage I: Specify one or more models to describe the process and perform preliminary identifiability and distinguishability tests before any data is collected in order to determine whether or not the parameters in the mathematical models can be uniquely identified and model structures can be distinguished from one another.

Stage II: Design experiments for model discrimination to select the best model representative of the process.

Stage III: Design experiments to improve the precision of the parameters within the best model to arrive at a statistically verified model formulation. The design criterion is to minimize the volume of the confidence region for the parameter estimates.

Once the best model is chosen, the parameters of the model are random variables with associated probability distributions. The uncertainty in these

parameters affects the predictions made with the models for design, optimization, and control. High degree of correlation may also exist within the model parameters. For example, Whiting and co-workers (85) developed a new sampling strategy, called equal probability sampling (EPS), to account for the correlations between thermodynamic model parameters and sample the parameter space more efficiently.

If a complex model is used to represent the physical phenomena, with high number of model parameters, it is time consuming and expensive to estimate all the parameters with high precision. This task proves especially difficult when there are nonlinearly related model parameters. In these cases, the number of experiments needed to be performed to identify all the model parameters is very costly. Therefore, sensitivity analysis may be performed to identify the most significant model parameters and the experimentation can be directed toward determination of these parameters alone, in order to reduce the cost and duration of the parameter estimation and model validation process. Parameter sensitivity analysis is used to quantify the effect of certain model parameters on the model output. Recently, Kontoravdi and co-workers (86) applied Sobol's method for global sensitivity analysis (GSA) to a mammalian cell culture producing monoclonal antibodies and identify the parameters that have a significant impact on the output. Each parameter space is sampled using a Sobol sequence, which yields no overlapping points to thoroughly investigate the entire range of parameter values. This method can be used as a precursor for experimental design to reduce the cost of experimentation. A review of sensitivity analysis techniques for chemical models was presented recently by Saltelli and coworkers (87).

5.4. Process Synthesis and Design. Process synthesis translates chemical synthesis into a chemical process. It encompasses the choice of various unit operations, how they are connected, and the optimization of the proposed plant. Process design activities start at this level and a flowsheet of the plant is generated according to these decisions and process simulators are used to predict mass and energy flows for the process. Commonly employed methodologies for selecting optimal process flowsheet configurations can be classified into four groups (88): (*1*) optimization-based approach; (*2*) hierarchical heuristic approach; (*3*) thermodynamic phenomena driven approach; (*4*) evolutionary methods.

The optimization approach to process synthesis involves (a) formulation of a complex flowsheet incorporating all the alternative process configurations, which are called superstructures and (b) identification of an optimal design configuration for a system to meet specified performance and cost objectives. Once the superstructure is known, combinatorial optimization methods, eg, mixed-integer nonlinear programming (MINLP) algorithms, can be used to solve the synthesis problem.

The literature in the area of process synthesis and process design under uncertainty has been concentrated on two focused application areas: (*1*) pollution prevention by design, and (*2*) designing for flexibility.

The earlier papers in synthesis under uncertainty with pollution prevention focus dealt with integrated environmental control systems for coal-based power systems. The work continued and extended to address synthesis problems in this area (88–90). Nuclear waste management posed a very hard synthesis problem

(91). The combinatorial, nonconvex nature of the problem was hard to solve even with deterministic optimization methods. Uncertainties associated with waste tank contents and models caused further problems and demanded new algorithms. The new stochastic annealing algorithm provided optimal and robust solution to this problem in the face of uncertainties with reasonable computational time (64). A multiobjective extension of this problem to include policy aspect was possible due to these new algorithms (92). Dantus and High (93) also used this new algorithm for waste minimization in methylene chloride process synthesis.

Acevedo and Pistikopoulos (94) also addressed process synthesis problems under uncertainty and presented a stochastic framework based on a two-state stochastic MINLP formulation for the maximization of a function comprising the expected value of the profit, operating, and fixed costs of the plant. Uncertain parameters were described by general probabilistic distribution functions and multiperiod formulations.

Increased environmental concerns in recent years have profoundly changed traditional process synthesis and design. If the environmental issues are addressed in early stages of design, there is greater flexibility and more opportunities to reduce environmental impacts at a lower cost. Recently, Diwekar (52) presented a generalized framework to address this problem. Figure 26 presents the different levels involved in this framework. The innermost level corresponds to models for process simulation. In this level, all possible process alternatives for a particular process are defined. Chemical process simulators, eg, AspenPlus (95), MultiBatchDS (96), or SuperPro (97), can be used for the innermost modeling. The second level corresponds to the sampling loop where the uncertainties can be specified in terms of probability distributions. Once probability

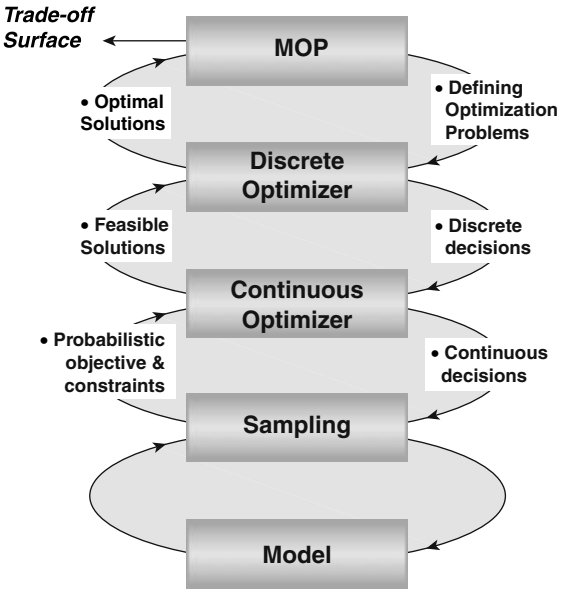

Fig. 26. Algorithmic framework for multiobjective optimization.

distributions are assigned to the uncertain variables, sampling techniques, eg, Monte Carlo sampling, HSS, or LHS can be used to perform the sampling operation from the multivariable uncertain parameter domain as discussed in the section on Sampling Techniques. The third level is the continuous optimizer that involves continuous decisions, eg, process design and operating conditions. In the fourth level, discrete decisions, such as chemical and process structural alternatives, are managed by mixed-integer programs. The outermost loop is the multiobjective programming loop where different objectives, eg, cost and environmental impacts, are considered and trade-off surfaces that are called Pareto optimal solutions are generated. Various applications of this approach, eg, hybrid fuel cell power plant design under uncertainty (1,98) and environmentally benign heterogeneous azeotropic distillation system design (68) have been presented. Kheawhom and Hirao (99) also presented a two layer algorithm for environmentally benign process synthesis under uncertainty. In the outer layer, the synthesis problem is represented by a multiobjective optimization problem considering the performances associated with design parameters. In the inner layer, the problem is expressed as a single-objective optimization problem taking in to account the operating performances in the presence of uncertainty. This algorithm was applied to a membrane-based toluene recovery process.

As mentioned earlier, it is essential to identify environmental impacts of a process earlier because the opportunity to overcome environmental problems in later stages of process development diminishes. However, in early design stages, there is high uncertainty in various economical, ecological, and technical process parameters. For this purpose, Hoffman and co-workers (100) proposed a new approach to select promising process alternatives in early stages of design. The method is based on approximating flowsheets by polynomial response surfaces with a lower complexity. A multiobjective optimization problem was solved for selecting a production process for hydrocyanic acid with 400 uncertain variables. Latin hypercube sampling technique was performed on the substituted response surface to obtain Pareto optimal solutions.

As stated earlier, process flexibility is an area that received significant attention, as it ensures that processes are operational and safe when exposed to variations in operating conditions. For example, for waste reduction in pharmaceutical industries (101,102) a discrete representation of waste loads assign probabilities to distinct waste scenarios. Since the explicit enumeration of all possible waste scenarios for numerous waste streams would lead to a massive amount of uncertain variables, a randomly selected sample is used to represent the uncertain space based on Monte Carlo sampling. Then a flexibility index is defined that measures the flexibility of a waste treatment policy to changing waste loads and superstructures are searched for recovery and treatment policies. Flexibility issues have also been addressed in process synthesis for heat exchanger network design (103,104), and synthesis of heat integrated distillation sequences (105). However, these papers use a scenario-based approach to represent uncertainties.

5.5. Process Operation. The aim of process operations is to optimally use capital, material, energy and information resources to produce desired chemical products in a reliable and flexible way while complying with environmental and safety regulations. Process operation involves activities ranging from

process control, monitoring and diagnosis, process planning, scheduling, and supply chain management. The following sections describe these various activities. The problems related to production planning, scheduling, and supply chain often involve discrete combinatory decisions and uncertainties. These problems belong to batch processing, which is generally used for the production of high value added, low volume products, eg, pharmaceuticals and specialty chemicals. Even though various sources of uncertainties exist in batch processing, most of the literature deal with demand uncertainties. To represent these uncertainties, sampling methods are used.

Process Planning, Management, and Scheduling. Most of the problems in management, scheduling, and planning include combinatorics (discrete choices and decisions) and uncertainties. Pekny (106) reviewed this area and provided algorithm structures that simultaneously consider combinatorial aspects and data uncertainty for industrial scale problems. The simulation-based optimization approach described in this article uses a customized mixed-integer linear programming solver to optimize process behavior together with a discrete event simulator to investigate the effect of uncertainty on the plans output from the optimizer. This procedure requires a biased sampling scheme to focus on critical events and avoid a large number of simulations that are not insightful.

Scheduling problems have been studied widely in the chemical engineering literature. Recently, Lin and Floudas (107) presented an overview of scheduling problems in multiproduct–multipurpose batch and continuous processes. The overall profitability of a process and the timely delivery of products highly depend on scheduling. Scheduling problems involve sequencing, assignment of tasks to equipment, and maintenance over a planning horizon, and inventory considerations of a process. Scheduling problems could be described conveniently using a resource-task equipment network. These problems are usually encountered in batch processing. In order to determine operating policies based on realistic production plans, the uncertain nature of processes must be addressed. In batch processing, uncertainties result from processing time fluctuations, equipment reliability or availability and demand. Two different approaches exist for scheduling problems under uncertainty: (*1*) reactive scheduling and (*2*) stochastic scheduling. In reactive scheduling, uncertainties are handled by adjusting the schedule when the uncertain parameters or unexpected events occur. Usually, heuristic approaches are used for schedule modifications. Whereas, in stochastic scheduling, the uncertainties are considered at the original scheduling stage and optimal and reliable schedules are found in the presence of uncertainty.

A scenario based approach is usually used, which comprises of all the possible future outcomes modeled by discrete probability distributions and the expected value of a performance index, eg, makespan or profit is optimized with respect to the scheduling decision variables (108). A scenario contains a discrete value for all uncertain variables within a given time interval and its associated probability. The number of scenarios increases exponentially with the number of uncertain variables, and this increases the problem size. Bassett and co-workers (109) presented a framework for addressing uncertainties by means of Monte Carlo sampling. Uncertain variable, eg, processing times and equipment downtimes, are sampled from their probability distributions and the reliability of meeting a certain due date is determined. Lee and Malone

(110) proposed a probabilistic approach based on the hybridization of Monte Carlo simulation and simulated annealing techniques to obtain a schedule able to handle uncertainties in parameters of batch process scheduling. This approach was also used to develop a flexible planning algorithm (111). Process planning involves the optimal selection of processes from among competing alternatives and timing of capacity expansions in a way that maximizes the net present value of the project over a planning horizon. Liu and Sahinidis (112) developed a two-stage stochastic program for process planning problems under uncertainty using a combination of Bender's decomposition and Monte Carlo sampling.

Supply Chain Management. Supply chain management takes the scheduling problems one step further and spans coordination of the management of multiple facilities and shipment of materials through an associated transportation network to customers. Supply chain management spans activities related to storage and movement of raw materials and products from the plant to the point of consumption. Due to changing market conditions and customer demands, it is critical for businesses to have an efficient and flexible supply chain. Various sources of short- and long-term uncertainties exist in these systems. Examples of short-term uncertainties are uncertainties in processing parameters, eg, processing times or yields or availability of equipment. On the other hand, long-term uncertainties include price fluctuations in raw material and final products and seasonal variations in demand, which occur over a longer period of time.

In literature, various sources of uncertainties are addressed in supply chain management. Gupta and Maranas (113) considered demand uncertainty in midterm planning of multisite supply chains. A stochastic programming-based approach was described to model the planning process as it reacts to demand realizations unfolding over time. Lababidi and co-workers (114) developed an optimization model to study the supply chain of a petrochemical company under uncertain operating and economic conditions. The objective function is based on optimizing the system resources by minimizing the total production costs and raw material procurement, as well as lost demand, backlog, transportation, and storage penalization. Uncertainties are considered in demands, market prices, raw material costs, and production yields. Multiple scenarios of an uncertain future, each with an associated probability of occurrence, were considered. It was found that uncertainties have a profound effect on the planning decisions of the petrochemical supply chain. Jung and co-workers (115) proposed the use of deterministic planning and scheduling models that incorporate safety stock levels as a means of accommodating demand uncertainties in routine operation by a Monte Carlo sampling technique. The problem of determining the safety stock level to use to meet a desired level of customer satisfaction is addressed using a simulation-based optimization approach. Wan and co-workers (116) extended the concept of simulation-based optimization by introducing a surrogate-based model together with domain reduction and incremental sampling to extract structure information from noisy simulation results and to optimize supply chain decisions. The idea behind a surrogate-based model is to fit a single surface for the whole decision space, and use this surface to perform optimization instead of the simulation model. This model is constructed using Latin hypercube sampling (LHS), domain reduction techniques to concentrate on the exploration of good regions, and support vector machines to extract structure information from noisy data.

Guillen and co-workers (117) presented a stochastic multiobjective optimization approach to obtain a trade-off between customer satisfaction and expected profit to be achieved in the short-term operation of chemical industry supply chains. A two-stage stochastic formulation is used, which considers the uncertainty associated with reactions to future demand and a set of Pareto optimal solutions are generated. This approach is aimed to provide decision support in making optimal offer proposals during negotiation process between customers and suppliers. The uncertainty associated with product demands and prices is represented by a set of scenarios with a given probability of occurrence and these scenarios are generated by performing Monte Carlo sampling.

Hung and co-workers (118) presented a new modeling approach based on an object-oriented architecture to handle supply chain configurations, operational decisions and policies, through the use of a generic supply chain node. The model provides a fully dynamic simulation of the supply chain and the effect of various uncertainties are evaluated through Monte Carlo simulation and other more efficient, sampling techniques based on quasi-Monte Carlo methods. The uncertain variables in supply chains are sampled from their respective probability distributions and the expected value of a performance indicator, eg, customer service level or average inventory is calculated.

Reliability. Because of increased competition worldwide, chemical plants need to operate with high process reliability to increase operational effectiveness and profits. System reliability and availability methods can be classified as measurement- and model-based methods (119). Measurement-based methods are expensive as they require building a real system or its prototype and taking measurements, and then analyzing the data statistically. In the context of process systems, at the design stage where the system or its prototype has not yet been built, the use of measurement techniques is not feasible. While at the operational stage, it can prove to be very expensive to inject faults into a real system to measure data. Model-based methods are much easier to use and are particularly useful at the design stage to screen lots of design alternatives without building the actual system. However, model-based methods are subject to model uncertainties, which propagate into RAM (reliability, availability, and maintainability) performance (120).

It has become important in recent years to address reliability issues at the conceptual design stage. It is also critical to increase the availability of the plant to save on lost production costs. The problem of including uncertainties in equipment availability at the design stage was addressed by Pistikopoulos and coworkers (121–124). Obtaining an optimal production schedule in the presence of equipment failure uncertainty for multiproduct/multipurpose batch plants is important for profitability and timely production. Sanmarti and co-workers (125) addressed this problem and introduced a schedule reliability index to identify robust schedules. This reliability index represents the discrete probability that a corresponding unit will be available to perform the next scheduled task based on the failure history and maintenance operations carried out on the unit. Production and maintenance schedules were determined simultaneously with this methodology. However, only the scenario-based approach is used for the probabilistic evaluation of reliability and availability in all these papers.

Genetic algorithms were applied to preventive maintenance optimization problems by Tan and Kramer (126). Their framework for preventive maintenance optimization combines Monte Carlo simulation with a genetic algorithm. This framework is suitable for handling uncertainties and nondeterministic objective functions.

Risk Analysis and Research Management. Risk and policy analysis involves uncertainty quantification and characterization using probability distributions and sampling. Since the results of the probabilistic analysis depend on the number of samples chosen, the choice of an efficient sampling technique becomes crucial. It is desirable to use a sampling technique that can predict the output probabilistic measure accurately with the minimum number of samples. Wang and co-workers (25) presented new sampling techniques based on the combination of HSS and LHS for the evaluation of health risk associated with exposure to hazardous materials. This sampling technique inherits the advantages of both HSS and LHS for superior efficiency.

Sampling techniques are also used for financial risk assessment in chemical process industries. For example, Bonfill and co-workers (127) presented a stochastic optimization approach to manage risk in short-term scheduling of multiproduct batch plants with demand uncertainties. A two-stage stochastic optimization model accounting for the maximization of the expected profit was used and this model was also extended to incorporate the availability of option contracts. To represent the demand uncertainty, independent scenarios were simulated by Monte Carlo sampling from the given probability distributions. A similar approach was applied for processing time uncertainties as well (128). Guillen and co-workers (129) developed a new strategy for integrating pricing decisions with the scheduling of batch plants to manage financial risk associated with demand uncertainty. The relationship between prices and demand have been modeled and forecasted and integrated into the scheduling model to determine simultaneously the prices and optimal schedule to maximize the profit. A sample average approximation (SAA) method was used to approximate the expected profit in the objective function.

Research management in general is related to research prioritization and reduction of uncertainties. A "value of research" methodology was proposed by Johnson and Diwekar (92) and Johnson and co-workers (130) for research management problems. This methodology tries to determine when imperfect information is acceptable, and where should the scarce resources be allocated to leverage the impact of these research efforts on the whole of its strategy. While reducing uncertainty is profitable, the time required to achieve a reduction tempers the benefit. This approach was applied recently to hybrid fuel cell power plants (131).

Robust Control. Control systems are used to keep the product specifications on target, to minimize deviations from the nominal process conditions, and maintaining the safe operation of the plant. Control system design involves the selection of input and output variables, the process model, appropriate type of controller, and adjustment of the controller tuning parameters. Model uncertainty and external disturbances are important concerns in designing control systems.

One of the most important criteria for designing a control system is to achieve robustness to these model uncertainties and disturbances. A probabilistic

approach has been used by Schaper and co-workers (132) to achieve robust process control. This probabilistic approach characterizes the model uncertainties by probability distributions and a statistical measure of disturbance rejection for the controller is incorporated into a robust control framework. The performance of the controller is then characterized by a probability measure for all situations between nominal and worst case conditions. Ratto and Paladino (133) considered uncertainties in process and kinetic parameters for nonideal controlled CSTRs and described a procedure to perform stability, sensitivity, and bifurcation analysis by a Monte Carlo method. This procedure is used to identify most probable stability regions and to design a robust control system. Li and co-workers (134) proposed a model predictive control strategy under chance constraints for robustness. Both the model and disturbance uncertainties were considered and assumed to be correlated multivariate stochastic variables. A stochastic program under joint probabilistic constraints was formulated and using the HSS technique, this problem was relaxed to a nonlinear programming problem.

In order to achieve robustness, parameter design methodology is also a widely used method. It is termed as an off-line quality control method for designing products and manufacturing processes that are robust in the face of uncontrollable variations popularized by Taguchi (135). The variables affecting a product's performance are classified into two groups: (*1*) design parameters whose nominal settings can be specified; (*2*) noise parameters that represent uncontrollable variations over a product's lifetime and across different units. In order to relate the noisy input parameters to the process output, two different approached could be used (*1*) physical experiments could be conducted by varying the input parameters over the noise space to generate a response surface, or (*2*) computational models could be developed. Monte Carlo methods are used for propagating the effects of input variability through a model and output variability is studied. A sample of input vectors is generated that is representative of the uncertainty distribution and outputs are evaluated at each of these samples.

The importance of sampling efficiency for generating these samples from the multivariate space for the formulation of a stochastic optimization problem for parameter design was emphasized in an earlier study by Diwekar and Rubin (90) for off-line quality control of a continuous stirred tank reactor. Latin hypercube sampling was used instead of Monte Carlo technique to reduce the required number of samples. Later, Kalagnanam and Diwekar (13) applied the HSS technique to this problem for further efficiency improvements exploiting the k-dimensional uniformity properties of this technique. Another study related to robust batch distillation column design using the HSS technique also illustrated the usefulness of this approach (14). Sahin and Diwekar (136) demonstrated efficiency of a new algorithm called better optimization of nonlinear uncertain systems (BONUS) for the same problem. Terwiesch and Agarwal (137) presented an optimization procedure to achieve robustness in batch reactor optimal control under parametric uncertainties. Probability distributions were used for the uncertain process parameters and the expectation of cost function for the entire parameter space was optimized.

Optimal Control. A challenging control problem that has received considerable attention in the literature is optimal control, where an optimal trajectory (future of action) for a control variable is computed by dynamic optimization, so

as to maximize/minimize a performance index, eg, cost, product yield, or time. To compute optimal control trajectories in the face of time-dependent uncertainties, a stochastic maximum principle formulation was developed based on real options theory and using a class of stochastic processes called *Ito processes* (51,138). By using this methodology, thermodynamic model uncertainties in batch distillation were characterized and stochastic optimal control profiles were computed (47,51,139). Sampling techniques and stochastic modeling approaches were used to characterize, quantify, and propagate uncertainties. This technique improved the process performance objectives significantly in various case studies.

In recent years, considering process design and control simultaneously has become important. Sakizlis and co-workers (140) provided a review of recent advances toward the integration of process design, process control, and process operability, where time-varying disturbances and uncertainties are considered. More recently, Pajula and Ritala (141) discussed the effects of measurement uncertainty on process performance and how it should be accounted for in the design of the control structure. For this purpose, dynamic scenarios were used and were each assigned a probability of occurrence using the knowledge gained from earlier experiences. This method was applied to a papermaking process where the effect of measurement uncertainty of fiber and filler consistency (concentration) on controller performance was studied. For batch separations and solvent recycling in the pharmaceutical industry, Ulas and Diwekar (142) presented a framework that couples product design, process design, and optimal control in the face of time-dependent uncertainties.

6. Conclusion and Future Trends

Sampling is an important element of uncertainty analysis, stochastic modeling, and optimization algorithms used for chemical process design, operation, and control. Sampling techniques are employed to sample probabilistic space of uncertain variables commonly encountered in these applications. Apart from uncertainty analysis, sampling also plays an important role in improving efficiency of discrete, stochastic, and multiobjective optimization algorithms. Sampling-based Monte Carlo methods are also an essential part of computational chemistry.

Monte Carlo sampling is the most commonly used sampling technique based on pseudorandom numbers. This sampling technique has probabilistic error bounds and requires large sample sizes to estimate the mean and standard deviation for an uncertain variable. Therefore, variance reduction techniques have been employed in order to increase efficiency. Importance sampling, LHS and HSS are examples of variance reduction techniques. Importance sampling is executed on the fact that some of the input random variables have more impact on the parameter being estimated than others. These values are sampled more frequently. On the other hand, LHS is a stratified sampling technique to reduce the sample size. Hammersley sequence sampling and its variants are efficient sampling techniques based on quasirandom numbers showing k-dimensional uniformity properties.

Because of increased environmental consciousness, traditional design methods should include objectives, eg, environmental and health impacts, risk, reliability and safety, as well as controllability and profitability, in earlier stages of process design. Sampling techniques have various applications during the life cycle of the plant, eg, chemical synthesis, process synthesis, and process operations, eg, management and planning, supply chains, scheduling, control and maintenance optimization, for better reliability. Environmental and financial risk management are other applications where sampling is a crucial step.

Future trends in sustainable process design require researchers to study the connections between industries and ecosystems, which are the complex networks of humans, plants, animals and the environment (2). The effects of hazardous chemicals and the activities of the chemical plants with the ecosystem need to be modeled for enhanced decision making. These multifaceted models have many steady-state and dynamic uncertainties and efficient sampling techniques will play an important role in analysis.

Furthermore in the future, the applicability of efficient sampling techniques, eg, HSS, needs to be examined for higher dimensional problems. There is a loss of uniformity observed in quasirandom sequences for higher dimensions. Leaping techniques (24,26) and combinations of quasirandom sequences are used to improve the HSS technique to make it applicable to higher dimensional problems and more work is needed in this area.

BIBLIOGRAPHY

1. U. M. Diwekar, *Environ. Sci. Technol.* **37**(23), 5432 (2003).
2. U. Diwekar, *Resources, Conservation Recycling* **44**(3), 215 (2005).
3. N. Metropolis and S. Ulam, *J. Am. Stat. Assoc.* **44**(247), 335 (1949).
4. D. H. Lehmer, *Proceedings of the 2nd Symposium on Large Scale Digital Calculating Machinery*, 141 (1949).
5. B. Wichmann and I. Hill, *Appl. Stat.* **31**, 188 (1982).
6. B. A. P. James, *J. Oper. Res. Soc.* **36**(6), 525 (1985).
7. B. L. Nelson, *Computers and Oper. Res.* **14**, 218 (1987).
8. B. L. Nelson, *Oper. Res.* **38**(6), 974 (1990).
9. J. R. Wilson, *Am. J. Math. Manag. Sci.* **3**, 121 (1983).
10. M. D. McKay, R. J. Beckman, and W. J. Conover, *Technometrics* **21**(2), 239 (1979).
11. R. L. Iman and W. J. Conover, *Commun. Stat.* **A17**, 1749 (1982).
12. E. Saliby, *J. Oper. Res. Soc.* **41**(12), 1133 (1990).
13. J. R. Kalagnanam and U. M. Diwekar, *Technometrics* **39**(3), 308 (1997).
14. U. M. Diwekar and J. R. Kalagnanam, *AIChE J.* **43**(2), 440 (1997).
15. D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, San Diego, Academic Press, New York, 1996.
16. D. E. Knuth, *The Art of Computer Programming Vol. 1: Fundamental Algorithms,* Addison-Wesley Publishing Co., New York, 1973.
17. D. A. Straub and I. E. Grossmann, *Comput. Chem. Eng.* **14**(9), 967 (1990).
18. F. P. Bernardo, E. N. Pistikopoulos, and P. M. Saraiva, *Comput. Chem. Eng.* **25**(1), 27 (2001).

19. M. Wendt, P. Li, and G. Wozny, *Ind. Eng. Chem. Res.* **41**(15), 3621 (2002).
20. V. R. Vasquez and W. B. Whiting, *Comput. Chem. Eng.* **23**(11–12), 1825 (2002).
21. H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM Publications, Philadelphia, 1992.
22. J. H. Halton, *Num. Math.* **2**, 84 (1960).
23. J. M. Hammersley, *Annal. NY Acad. Sci.* **86**, 844 (1960).
24. K. Subramanyan and U. M. Diwekar, *User Manual for Fortran-Based Stochastic Sampling Code*, 2006.
25. R. Y. Wang, U. M. Diwekar, and C. E. G. Padro, *Environ. Prog.* **23**(2), 141 (2004).
26. L. Kocis and W. Whiten, *ACM Trans. Math. Software* **23**(2), 266 (1997).
27. P. Hellekalek, *Proceedings of the 12th Workshop on Parallel and Distributed Simulation IEEE*, 1998, pp. 82–89.
28. M. Mascagni, in R. Schreiber, M. Heath, and A. Ranade, eds., *Algorithms for Parallel Processing*, Springer Verlag, New York, 1997, pp. 277–288.
29. M. Mascagni, in J. Dongarra, I. Foster, F. Fox, W. Gropp, K. Kennedy, L. Torcson, and A. White, eds., *Sourcebook on Parallel Computing*, Morgan Kaufman Publishers, San Francisco, 2003, pp. 249–258.
30. W. C. Schmid and A. Uhl, *Math. Comp. Simulation* **55**(1–3), 249 (2001).
31. T. Bayes, *Philos. Trans. R. Soc. London* **53**, 370 (1763).
32. A. Meel, L. M. O'Neill, W. D. Seider, U. Oktem, and N. Keren, *AIChE Spring National Meeting*, Paper 106d (2006).
33. N. Mehranbod, M. Soroush, and C. Panjapornpon, *J. Process Control* **15**(3), 321 (2005).
34. Y. Bard, *Nonlinear Parameter Estimation*, Academic Press, New York, 1974, pp. 72–74.
35. S. Geman and D. Geman, *IEEE Trans. Pattn. Anal. Mach. Intell.* **6**, 721 (1984).
36. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
37. W. K. Hastings, *Biometrika* **57**, 97 (1970).
38. S. P. Brooks, *The Statistician* **47**(Part 1), 69 (1998).
39. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, Great Britain, 1996.
40. W. G. Madow and L. H. Madow, *Ann. Math. Stat.* **15**(1), 1 (1944).
41. S. K. Thompson, *J. Am. Stat. Assoc.* **85**(412), 1050 (1990).
42. C. Laurent and J. F. Etard, *Rev. D'Epidemiol. de Sante Publique* **53**(1), 43 (2005).
43. P. Turk and J. J. Borkowski, *Environ. Ecol. Statistics* **12**(1), 55 (2005).
44. T. M. F. Smith, *J. R. Stat. Soc. Ser. A (General)* **146**(4), 394 (1983).
45. A. S. Whittemore and J. Halpern, *Stat. Med.* **16**(1–3), 153 (1997).
46. A. Chaudhuri, A. K. Adhikary, and S. Dihidar, *Metrika* **52**(2), 115 (2000).
47. S. Ulas and U. M. Diwekar, *Comput. Chem. Eng.* **28**(11), 2245 (2004).
48. U. M. Diwekar and E. S. Rubin, *Comput. Chem. Eng.* **15**(2), 105 (1991).
49. K.-J. Kim and U. M. Diwekar, *Ind. Eng. Chem. Res.* **41**(5), 1276 (2002).
50. K.-J. Kim and U. M. Diwekar, *Ind. Eng. Chem. Res.* **41**(5), 1285 (2002).
51. V. Rico-Ramirez, U. M. Diwekar, and B. Morel, *Comput. Chem. Eng.* **27**(12), 1867 (2003).
52. U. M. Diwekar, *Introduction to Applied Optimization*, Kluwer Academic Publishers, Boston, Mass., 2003.
53. A. K. Dixit and R. S. Pindyck, *Investment under Uncertainty*, Princeton University Press, Princeton, N.J., 1994.
54. P. J. M. VanLaarhoven and E. H. Aarts, *Simulated Annealing Theory and Applications*, D. Reidel Publishing Co., Holland, 1987.
55. L. A. Painton and U. M. Diwekar, *Eur. J. Oper. Res.* **83**, 489 (1995).

56. K.-J. Kim and U. M. Diwekar, *IIE Trans.* **34**(9), 761 (2002).
57. U. M. Diwekar and W. Xu, *Ind. Eng. Chem. Res.* **44**(18), 7132 (2005).
58. J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Series in Operations Research, Springer-Verlag, New York, Inc., 1997.
59. J. R. Birge and F. Louveaux, *Eur. J. Oper. Res.* **34**, 384 (1988).
60. G. B. Dantzig and G. Infanger, in C. Brezinski and U. Kulisch, eds., *Computational and Applied Mathematics I-Algorithms and Theory*, Proceedings of the 13th IMCAS World Congress, Dublin, Ireland, 1991, pp. 111–120.
61. J. L. Higle and S. Sen, *Math. Oper. Res.* **16**(3), 650 (1991).
62. J. Wei and M. J. Realff, *Comput. Chem. Eng.* **28**(3), 333 (2004).
63. P. D. Chaudhuri and U. M. Diwekar, *AIChE J.* **42**, 742 (1996).
64. P. Chaudhuri and U. Diwekar, *AIChE J.* **45**(8), 1671 (1999).
65. S. Sambarey and U. M. Diwekar, *Grace Hopper Celebration of Women in Computing*, San Diego, Calif., 2006.
66. C. Hwang, S. Paidy, and K. Yoon, *Comput. Operations Res.* **7**, 5 (1980).
67. Y. Fu and U. M. Diwekar, *Ann. Oper. Res.* **132**, 109 (2004).
68. W. Xu and U. M. Diwekar, *Ind. Eng. Chem. Res.* **44**(11), 4061 (2005).
69. A. Leach, *Molecular Modeling Principles and Applications*, Prentice Hall, Pearson Education Limited, Great Britain, 2001.
70. J. I. Siepmann and D. Frenkel, *Mol. Phys.* **75**(1), 59 (1992).
71. T. J. H. Vlugt, M. G. Martin, B. Smit, J. I. Siepmann, and R. Krishna, *Mol. Phys.* **94**(4), 727 (1998).
72. M. G. Martin and J. I. Siepmann, *J. Phys. Chem. B* **103**(21), 4508 (1999).
73. Z. Chen and F. A. Escobedo, *J. Chem. Phys.* **113**(24), 11382 (2000).
74. Y. Okamoto, *J. Mol. Graphics Modeling* **22**(5), 425 (2004).
75. A. Mitsutake, Y. Sugita, and Y. Okamoto, *Biopolymers* **60**(2), 96 (2001).
76. S. Ulas and U. M. Diwekar, *Molecular Simulation* (accepted, invited paper for the special issue of Frontiers in Molecular Simulations), 2006b.
77. K. G. Joback and R. C. Reid, *Chem. Eng. Commun.* **57**(1–6), 233 (1987).
78. H. K. Hansen, P. Rasmussen, A. Fredenslund, M. Schiller, and J. Gmehling, *Ind. Eng. Chem. Res.* **30**(10), 2352 (1991).
79. P. M. Harper, G. Rafiqul, P. Kolar, and T. Ishikawa, *Fluid Phase Equil.* **158–160**, 337 (1999).
80. C. D. Maranas, *AIChE J.* **43**(5), 1250 (1997).
81. M. C. Tayal and U. M. Diwekar, *AIChE J.* **47**(3), 609 (2001).
82. K.-J. Kim and U. M. Diwekar, *Ind. Eng. Chem. Res.* **41**(18), 4479 (2002).
83. W. Xu and U. M. Diwekar, *Ind. Eng. Chem. Res.* **44**(18), 7138 (2005).
84. S. P. Asprey and S. Macchietto, *Comput. Chem. Eng.* **24**(2–7), 1261.
85. W. B. Whiting, V. R. Vasquez, and M. M. Meerschaert, *Fluid Phase Equil.* **158–160**, 627 (1999).
86. C. Kontoravdi, S. P. Asprey, E. N. Pistikopoulos, and A. Mantalaris, *Biotechnol. Prog.* **21**(4), 1128 (2005).
87. A. Saltelli, M. Ratto, and F. Campolongo, *Chem. Rev.* **105**(7), 2811 (2005).
88. U. M. Diwekar, E. S. Rubin, and H. C. Frey, *Energy Conv. Management* **38**, 1725 (1997).
89. U. M. Diwekar, *AIChE Symp. Ser. Pollution Prevention through Process and Product Mod.* **90**(303), 168 (1995).
90. U. M. Diwekar and E. S. Rubin, *Ind. Eng. Chem. Res.* **33**(2), 292 (1994).
91. V. Narayan, U. M. Diwekar, and M. Hoza, *Ind. Eng. Chem. Res.* **35**(10), 3519 (1996).
92. T. L. Johnson and U. M. Diwekar, in M. F. Malone, J. A. Tainham, and B. Carnahan, eds., *Foundations of Computer-Aided Design*, *AIChE Symp. Ser.* **96**, 454 (2000).
93. M. M. Dantus and K. A. High, *Comput. Chem. Eng.* **23**(10), 1493 (1999).

94. J. Acevedo and E. N. Pistikopoulos, *Comput. Chem. Eng.* **22**, 647 (1998).
95. Aspen Tech, AspenTech. Aspen Plus 12.1 User Guide. Cambridge, Mass., 2003.
96. U. M. Diwekar, *Comput. Appl. Chem. Eng. Educ.* **4**(4), 275 (1996).
97. T. Shanklin, K. Roper, P. K. Yegneswaran, and M. Marten, *Biotechnol. Bioeng.* **72**(4), 483 (2001).
98. K. Subramanyan, U. Diwekar, and A. Goyal, *J. Power Sources* **132**(1–2), 99 (2004).
99. S. Kheawhom and M. Hirao, *Comput. Chem. Eng.* **28**(9), 1715 (2004).
100. V. H. Hoffmann, G. J. McRae, and K. Hungerbuhler, *Ind. Eng. Chem. Res.* **43**(15), 4337 (2004).
101. A. A. Linninger, A. Chakraborty, and R. D. Colberg, *Comput. Chem. Eng.* **24**(2–7), 1043 (2000).
102. A. A. Linninger and A. Chakraborty, *Comput. Chem. Eng.* **25**(4–6), 675 (2001).
103. L. Tantimuratha and A. C. Kokossis, *Annal. Oper. Res.* **132**, 277 (2004).
104. Z. N. Pintaric and Z. Kravanja, *Comput. Chem. Eng.* **28**(6–7), 1105 (2004).
105. G. E. Paules and C. A. Floudas, *Comput. Chem. Eng.* **16**, 189 (1992).
106. J. F. Pekny, *Comput. Chem. Eng.* **26**(2), 239 (2002).
107. X. Lin, S. L. Janak, and C. A. Floudas, FOCAPO 2003 Special issue. *Comput. Chem. Eng.* **28**(6–7), 1069 (2004).
108. M. G. Ierapetritou and E. N. Pistikopoulos, *Ind. Eng. Chem. Res.* **35**(3), 772 (1996).
109. M. H. Bassett, J. F. Pekny, and G. V. Reklaitis, *Comput. Chem. Eng.* **21**(Suppl. 1), S1203 (1997).
110. Y. G. Lee and M. F. Malone, *Inter. J. Prod. Res.* **39**(4), 603 (2001).
111. Y. G. Lee and M. F. Malone, *Ind. Eng. Chem. Res.* **40**(6), 1507 (2001).
112. M. L. Liu and Sahinidis, *Ind. Eng. Chem. Res.* **35**(11), 4154 (1996).
113. A. Gupta and C. D. Maranas, *Comput. Chem. Eng.* **27**(8–9), 1219 (2003).
114. H. M. S. Lababidi, M. A. Ahmed, I. M. Alatiqi, and A. F. Al-Enzi, *Ind. Eng. Chem. Res.* **43**(1), 63 (2004).
115. J. Y. Jung, G. Blau, J. F. Pekny, G. V. Reklaitis, and D. Eversdyk, *Comput. Chem. Eng.* **28**(10), 2087 (2004).
116. X. Wan, J. F. Pekny, and G. V. Reklaitis, *Comput. Chem. Eng.* **29**(6), 1317 (2005).
117. G. Guillen, F. D. Mele, M. J. Bagajewicz, A. Espuna, and L. Puigjaner, *Chem. Eng. Sci.* **60**(6), 1535 (2005).
118. W. Y. Hung, N. J. Samsatli, and N. Shah, *Eur. J. Oper. Res.* **169**(3), 1064 (2006).
119. A. Sathaye, S. Ramani, and K. S. Trivedi, *Proceedings of International Workshop on Fault-Tolerant Control and Computing* (FTCC-1), 2000.
120. H. Goel, *Integrating Reliability, Availability and Maintainability (RAM) in Conceptual Process Design, An Optimization Approach*, Delft University Press, Delft, The Netherlands, 2004.
121. E. N. Pistikopoulos, T. V. Thomaidis, A. Melin, and M. G. Ierapetritou, *Comput. Chem. Eng.* **20**, S1209 (1996).
122. E. N. Pistikopoulos, C. G. Vassiliadis, and L. G. Papageorgiou, *Comput. Chem. Eng.* **24**, 203 (2000).
123. E. N. Pistikopoulos, C. G. Vassiliadis, J. Arvela, and L. G. Papageorgiou, *Ind. Eng. Chem. Res.* **40**, 3195 (2001).
124. C. G. Vassiliadis and E. N. Pistikopoulos, *Comput. Chem. Eng.* **23**, S555 (1999).
125. E. Sanmarti, A. Espuna, and L. Puigjaner, *Comput. Chem. Eng.* **21**, 1157 (1997).
126. J. S. Tan and M. A. Kramer, *Comput. Chem. Eng.* **21**, 1451 (1997).
127. A. Bonfill, M. Bagajewicz, A. Espuna, and L. Puigjaner, *Ind. Eng. Chem. Res.* **43**(3), 741 (2004).
128. A. Bonfill, A. Espuna, and L. Puigjaner, *Ind. Eng. Chem. Res.* **44**(5), 1524 (2005).
129. G. Guillen, M. Bagajewicz, S. E. Sequeira, A. Espuna, and L. Puigjaner, *Ind. Eng. Chem. Res.* **44**(3), 557 (2005).

130. T. L. Johnson and U. M. Diwekar, *J. Multi-Criteria Decision Anal.* **10**, 87 (2001).
131. K. Subramanyan and U. M. Diwekar, *Ind. Eng. Chem. Res.* **45**(2), 681 (2006).
132. C. D. Schaper, D. E. Seborg, and D. A. Mellichamp, *Ind. Eng. Chem. Res.* **31**(7), 1694 (1992).
133. M. Ratto and O. Paladino, *Chem. Eng. J.* **79**(1), 13 (2000).
134. P. Li, M. Wendt, and G. Wozny, *Comput. Chem. Eng.* **24**(2–7), 829 (2000).
135. G. Taguchi and Y. Wu, *Introduction to Off-line Quality Control*, Nagoya, Japan, Central Japan Quality Control Association, 1980.
136. K. Sahin and U. Diwekar, *Ann. Oper. Res.* **132**, 47 (2004).
137. P. Terwiesch and M. Agarwal, *Chem. Eng. Commun.* **131**, 33 (1995).
138. V. Rico-Ramirez and U. M. Diwekar, *Comput. Chem. Eng.* **28**(12), 2845 (2004).
139. S. Ulas, U. M. Diwekar, and M. A. Stadtherr, *Comput. Chem. Eng.* **29**(8), 1805 (2005).
140. V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos, *Comput. Chem. Eng.* **28**(10), 2069 (2004).
141. E. Pajula and R. Ritala, *Chem. Eng. Proc.* **45**(4), 312 (2006).
142. S. Ulas and U. M. Diwekar, *Chem. Eng. Sci.* **61**(6), 2001 (2006).

URMILA M. DIWEKAR
University of Illinois at Chicago, Vishwamitra Research Institute
SAADET ULAS
University of Illinois at Chicogo, Vishwamitra Research Institute