Process Synthesis under Uncertainty: A Penalty Function Approach

Prosenjit D. Chaudhuri and Urmila M. Diwekar
Dept. of Civil & Environmental Engineering and Engineering and Public Policy,
Carnegie Mellon University, Pittsburgh, PA 15213

With the growing environmental concern, it is necessary to improve process simulation and develop design tools to account for environmental factors in the synthesis of large-scale chemical processes. A major obstacle in tackling this problem is uncertainties in some of the technical and economic parameters, which lead to uncertainties in design, plant performance, and cost estimates. Further, a conceptual process design involves the identification of an optimal flowsheet structure from many alternatives stituting the "superstructure." Synthesis and optimization of large-scale processes involving uncertainties often require considerable computational effort. A novel algorithm presented here is based on simulated annealing for the process synthesis of large-scale flowsheets having several configurations and considers uncertainties in the process design systematically. This new "stochastic annealing algorithm," provides an efficient approach to stochastic synthesis problems by incorporating a penalty term in the objective function and balances the trade-off between accuracy and efficiency based on the annealing temperature. It has been used to study a benchmark synthesis problem in the HDA process. Savings of up to 80% in CPU time has been achieved without significant loss of solution precision with stochastic annealing, compared to simulated annealing with a fixed sample size. It can be applied to analyze efficiently any complex process flowsheet and provide valuable insights into process feasibility based on optimal design, plant performance, and uncertainty issues.

Introduction

Computer-aided process design plays an important role in the design of new processes and the analysis of existing processes. The design of new processes is, however, complicated by technical and economic uncertainties, which lead to uncertainties in the prediction of plant performance and overall plant economics. An example where such technical and economic uncertainties occur and are not treated or characterized rigorously, is in the design of integrated environmental control processes for advanced power systems (Diwekar et al., 1992b). Since the conceptual design of any chemical process involves the identification of possible flowsheet configurations, design methods must also address the issues of process synthesis under uncertainty, as it has important implications on process viability, and other quality measures such as controllability, safety, and environmental compliance.

Correspondence concerning this article should be addressed to U. M. Diwekar.

Conventional simulation models are based on a steady-state deterministic framework and do not handle uncertainties in a systematic manner. The necessity to analyze uncertainties is greater in the context of emerging processes, since for these processes the available performance data are scant and the technical and economic parameters are not well established. A systematic framework to analyze uncertainties is a key step in this regard and promises to overcome some of the difficulties encountered with deterministic simulators. Furthermore, although design under uncertainty has received considerable attention in the past in the chemical engineering literature (Pistkopoulos and Grossmann, 1988; Straub and Grossmann, 1990; Varvarezos et al., 1993; Ciric and Huchette, 1993), a generalized framework for analyzing uncertainty systematically, however, has recently been developed around a chemical process simulator (Diwekar and Rubin, 1991).

One of the main goals in synthesis problems is to establish

methodologies for selecting optimal flowsheet configurations. Approaches to process synthesis problems essentially fall under the following areas:

1. Thermodynamic approach (Linnhoff, 1981)
2. Evolutionary methods (Nishida et al., 1981)
3. Hierarchical approach, based on intuition and engineering judgment (Douglas, 1985)
4. Optimization approach based on mathematical programming techniques (Grossmann, 1985).

In general, mathematical programming techniques in process synthesis involves (a) formulation of a conceptual flowsheet incorporating all the alternative process configurations (superstructure) and (b) identification of an optimal design configuration based on optimal structural topology and the optimal parameter-level settings for a system to meet specified performance and cost objectives (Grossmann, 1990). Once the superstructure is known, mixed integer nonlinear programming (MINLP) algorithms can be used to solve the synthesis problem. These algorithms are based on the alternating sequence of nonlinear programs (NLPs) and mixed integer linear programs (MILPs). Over the years, significant advances in MINLP algorithms have lead to the development of equation-oriented software, such as APROS (Paules and Floudas, 1989), DICOPT++ (Kocis and Grossmann, 1989), and PROSYN (Kravanja and Grossmann, 1990). These packages have some practical limitations, as it may be too laborious and cumbersome to analyze complex chemical processes.

Sequential-modular deterministic simulators, such as ASPEN/PRO-II, incorporate detailed process models and have been developed to simulate complex processes. However, they do not possess any synthesis capability. A synthesis capability incorporating MINLP optimization techniques has been developed around the public version of ASPEN, but even then it lacked the ability to synthesize processes under uncertainty (Diwekar et al., 1992a). Optimization under uncertainty essentially falls into two classes of problems: (1) stochastic optimization; (2) stochastic programming. In stochastic optimization, at each optimization iteration, some probabilistic representation of the objective function and constraints is optimized. On the other hand, stochastic programming involves the solution of a deterministic optimization for each scenario, yielding a probabilistic representation of optimal solutions. A methodology for stochastic optimization and stochastic programming capability built around a deterministic simulator has been outlined recently (Diwekar, 1994).

An alternative to the solution of the superstructure using MINLP techniques is simulated annealing. The advantages of simulated annealing in chemical process synthesis applications are that it is not a derivative-based method and can handle large discontinuities in the solution space.

Simulated annealing, in recent years, has been successfully applied to the design of heat exchanger and pipeline networks (Dolan et al., 1989) and in the scheduling of batch processes (Das et al., 1990). However, unlike MINLP, it has not been incorporated as a design mechanism in synthesis problems. Further, an algorithm based on simulated annealing, developed to handle uncertainties in a systematic and efficient manner has not been applied to the study of large-scale flowsheets.

This article therefore presents a new variant of simulated annealing, which has been modified to handle uncertainties in the form of probabilistic distributions and can be extended to process synthesis applications. Since stochastic analysis increases computational time owing to the large number of samples required to arrive at a more definitive probabilistic value of the objective function, another goal of this algorithm is to achieve the trade-off between accuracy and efficiency based on the annealing temperature. We have developed an algorithm, referred as "stochastic annealing," that can select the optimal number of samples and obtain the desired optimal solution without significant loss of accuracy. This endeavor is promising in the context of large-scale flowsheets in determining the optimal design configuration as well as handling uncertainties in a manner that is computationally efficient, yet predicts probabilistic estimates of plant performance, design, and economics accurately.

Stochastic Modeling and Synthesis under Uncertainty

The goal of an optimization problem is to determine the set of decision variables θ that optimize some aspect of the deterministic model represented by objective function Z, subject to equality constraints h and inequality constraints g. A mathematical formulation of the problem is given by

$$\underset{\theta}{\text{Optimize}} \quad Z = z(\theta,x) \tag{1}$$

subject to

$$h(\theta,x) = 0 \tag{2}$$

$$g(\theta,x) \le 0 \tag{3}$$

where θ is a decision variable vector and x is a vector of model parameters.

In many cases, the need for realistic models necessitates an uncertainty analysis owing to uncertainties associated with some of the input parameters in the model. A general approach to a wide variety of problems involving uncertainties is to assign probability distributions to the various uncertain parameters. Hence, a generalized stochastic optimization problem where the decision variables and uncertain parameters are separable is as follows:

$$\underset{\theta}{\text{Optimize}} \quad Z = P_1[z(\theta,x,u)] \tag{4}$$

subject to

$$P_2[h(\theta,x,u)] = \alpha_1 \tag{5}$$

$$P_3[g(\theta,x,u)] \le \alpha_2, \tag{6}$$

where u is a vector of uncertain parameters and P_x represents a probabilistic function. The vector of uncertain parameters (u) are chosen based on the amount of information available (Diwekar and Rubin, 1991). For an expected value minimization of a function Z with a cumulative probability distribution p, this results in:

$$\text{Minimize} \quad \bar{z} = E(Z) = \int_0^1 z \; dp. \tag{7}$$

For N_{samp} samples, the expected value can be obtained by sampling from the distribution:

$$E(Z) = \frac{\sum_1^{N_{samp}} z}{N_{samp}}. \tag{8}$$

On the other hand, if one seeks to minimize the variance, then the probabilistic functional case for variance minimization becomes:

$$\text{Minimize} \quad \sigma^2 = \int_0^1 (z_i - \bar{z})^2 \, dp \tag{9}$$

$$= \frac{\sum_1^{N_{samp}} (z_i - \bar{z})^2}{N_{samp} - 1}. \tag{10}$$

It is apparent from the previous equations that unlike a deterministic optimization problem, a stochastic optimization problem treats a probabilistic function of the objective function and constraints. In stochastic optimization, therefore, the stochastic modeler assigns the probabilistic distributions to the input parameter u, then uses a sampling technique to generate the specified number of samples (N_{samp}) and passes the sampled value of each parameter to the model. For each model run, the objective function and constraints are collected. The simulation is then repeated for a new set of samples selected from the probability distributions. Finally, when all the samples have gone through the cycle, the stochastic modeler analyzes the output and finds the probabilistic function of the objective function and constraints, which is passed on to the optimizer. Since at each optimization iteration stage, one needs to run the stochastic model with a large number of samples to calculate the probabilistic functions, the computational intensity in stochastic optimization is large.

There are many ways of sampling from probability distributions, of which the best known and more commonly used is Monte Carlo sampling. In crude Monte Carlo sampling, a value is drawn from a distribution at random for each input parameter. In most cases, the primary objective is to produce a more uniform distribution of points in the parameter space; then systematic or *stratified* sampling techniques become more appealing. In stratified sampling, the sampling space is segmented into intervals and the input values are obtained by sampling separately from each interval, instead of from the entire distribution. Latin hypercube sampling (LHS), which represents one version of stratified sampling (Iman and Shortencarier, 1984), is used in our analysis.

Implication of Sample Size in Simulation Experiments

Simulation runs involving uncertain parameters in the model are computationally intensive due to the large number of samples that need to be propagated through the model. One of the primary concerns in simulation runs involving uncertain parameters is therefore the sample size one needs to choose to obtain precisely an estimate for a statistical parameter, such as the mean or the standard deviation, without wasting too much CPU time. Monte Carlo techniques have the advantage of estimating the precision of the estimated parameters based on a particular sample size. This is because one can apply standard statistical techniques to analyze the output from a Monte Carlo run, as the sampled values of each output variable is a "random" sample from a true probability distribution of that variable. In most cases, the number of sampling runs depends on the cost and objective of the run. Furthermore, the number of runs also depends upon the type of application. For a comprehensive discussion on the number of sample runs required to build reliable models, the reader is referred to Morgan and Henrion (1990).

In almost all stochastic optimization problems, the major bottleneck is the computational time for generating and evaluating probabilistic functions of the objective function and constraints. For a given number of samples (m) of a random variable (Y), the estimate for the mean or expected value ($\bar{y}$) and the unbiased estimator for standard deviation (s) can be obtained from classic statistics (Milton and Arnold, 1990).

The accuracy of the estimates for the actual mean (μ) and the actual standard deviation (σ) is particularly important to obtain realistic estimates of any performance or economic parameter. However, as stated earlier, this accuracy is dependent on the number of samples. The number of samples required for a given accuracy in a stochastic optimization problem depends upon several factors, such as the type of uncertainty and the point values of the decision variables (Painton and Diwekar, 1995). The idea of interval estimation in statistics enables us to construct a confidence interval around the mean or standard deviation and allows us to make inferences on the mean or standard deviation of a distribution. It seems plausible, therefore, to use this idea of interval estimation to select the number of samples to obtain more accurate estimates of any statistical parameter without wasting unnecessary CPU time. In our approach, we have used this notion of a confidence interval to estimate the number of samples required to obtain precise estimates of a statistical parameter, such as the mean or the standard deviation.

Confidence Interval and the Bandwidth

The sampling distribution of $\bar{y}$ depends on the underlying distribution of Y. If Y is normally distributed, then $\bar{y}$ follows a t distribution and approaches a normal distribution as m becomes larger. Further, no matter what the distribution of Y might be, for large samples, the distribution $\bar{y}$ approaches a normal distribution with mean μ and standard deviation $\sigma/\sqrt{m}$. This is a result of the central limit theorem, and enables one to represent the uncertainty in the estimated mean $\bar{y}$. Since s is an unbiased estimator for σ, one can estimate the standard deviation of the sample mean. For Z, a random variable with unit normal distribution and c the corresponding deviation such that the range $(-c, c)$ encloses α probability, the confidence interval for μ, the mean of Y is $\bar{y} - c(s/\sqrt{m})$, $\bar{y} + c(s/\sqrt{m})$. This, however is not an accurate estimate if m is small.

On the other hand, if one chooses to minimize the standard deviation (s) to arrive at a less "uncertain" expected value of the mean of an output result, estimates of the upper and lower bounds of the standard deviation can be obtained from the chi-square distribution. The underlying assumption in these estimates is that the sampling is performed from a normal distribution.

To illustrate how interval estimation can be used to select the sample size, consider a simple probabilistic function $f(x_1,$

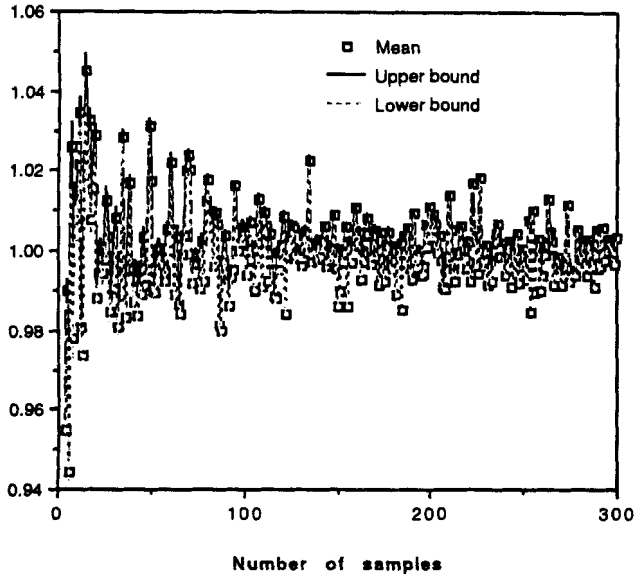

Figure 1. Mean and its bounds as a function of the number of samples for Monte Carlo runs.

$x_2) = x_1 \times x_2$ of two variables x_1 [uniform (0.9, 1.1)] and x_2 [normal (0.8, 1.2)]. For a given number of samples, one can construct upper and lower bounds for a 95% confidence interval around a parameter, which gives a measure of the precision in the estimate for that parameter. Figure 1 shows the estimates of the mean for $f(x_1, x_2)$ vs. the number of samples for crude Monte Carlo runs. Comparing this with Figure 2, which shows the estimate of the mean of $f(x_1, x_2)$ vs. the number of samples, for Latin hypercube sampling, it can be observed that Latin hypercube sampling requires a fewer number of samples to obtain precise estimates of the mean. The upper and lower bounds (shown by the solid and broken lines, respectively) for a 95% confidence interval for LHS indicate that there is no significant change in the width of the interval after 150 samples. This bound is truly an overestimate for Latin hypercube sampling; nevertheless, it allows one to perceive the number of samples required for Latin hypercube sampling based on a confidence interval around the mean. Similar results are obtained for the estimates of the standard deviation. Figure 3 indicates that a large number of samples are required to obtain precise estimates of the standard deviation for Monte Carlo runs compared to Latin hypercube sampling (Figure 4). Again, although the bounds for the estimate of the standard deviation is an overestimate for Latin hypercube sampling, they show a similar trend, as in the case of the estimate of the mean, namely, that the width of the confidence interval does not show a significant change above a certain number of samples (in this case, 250). This leads us to believe that one can use the confidence interval to select the number of samples in simulation runs.

It seems clear from the preceding discussion that for an especially small number of samples, the interval estimation for the mean or the standard deviation discussed before, is an approximation. Currently, a novel approach based on the concept of fractal dimensions to estimate the error bandwidth accurately is being investigated (Diwekar, 1994).

Combinatorial Optimization and Synthesis under Uncertainty

Process synthesis may involve a large number of discrete configurations of a given flowsheet. Such problems involving uncertainties, coupled with combinatorial optimization can make the solution space highly nonlinear and discontinuous and sometimes impossible for a search algorithm to converge on the optimum. An example of such a system that can cause such a "combinatorial explosion" is a simple model of a Brayton cycle power plant (Painton and Diwekar, 1994). This sys-

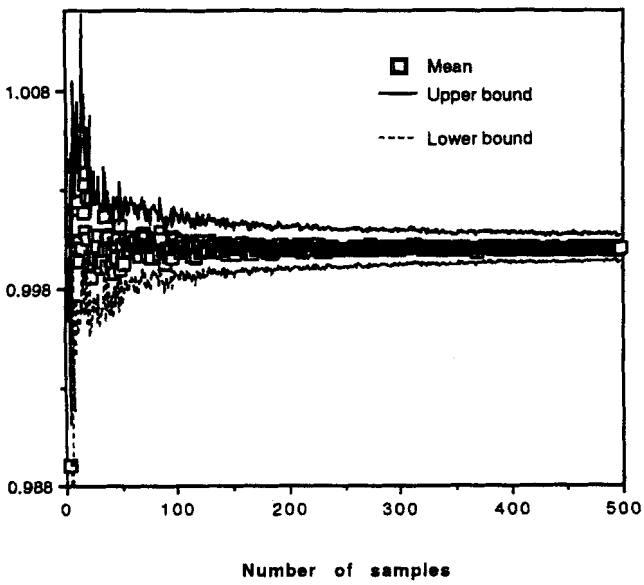

Figure 2. Mean and its bounds as a function of the number of samples for Latin Hypercube sampling.

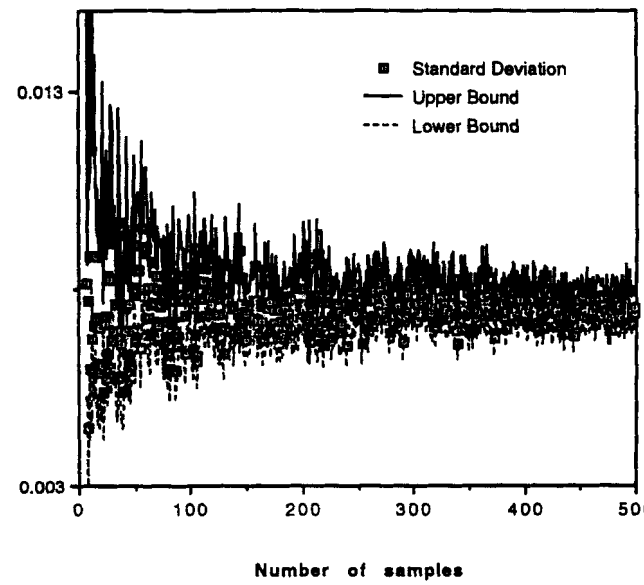

Figure 3. Standard deviation and its bounds as a function of the number of samples for Monte Carlo runs.

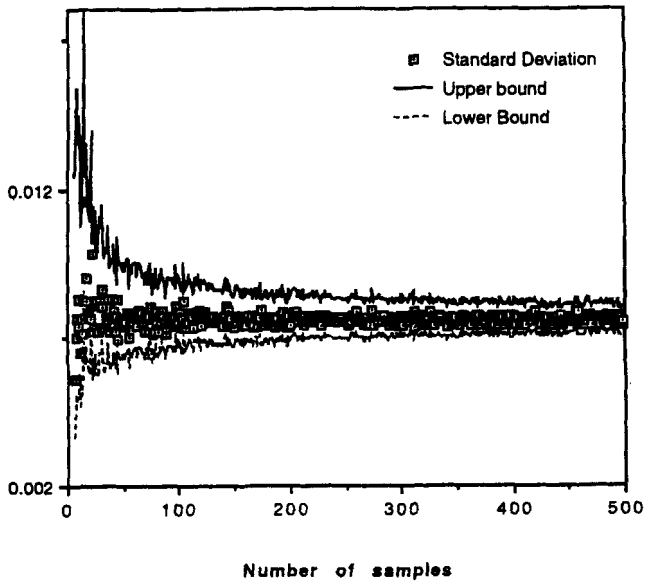

Figure 4. Standard deviation and its bounds as a function of the number of samples for Latin Hypercube sampling.

tem has been shown to yield a solution space comprising 10.2 million combinations. In the past decade, simulated annealing has been known to solve such combinatorial optimization problems and has been applied in widely different fields (Rutenbar, 1989). However, to the best of our knowledge, the problem of a systematic treatment of uncertainty using simulated annealing has never been addressed in the engineering literature. In the subsequent sections, we present the standard simulated annealing algorithm and its new variant, the stochastic annealing algorithm, which can treat a probabilistic objective function systematically as well as select the optimum number of samples based on the error bandwidth.

Simulated annealing algorithm

Simulated annealing is a heuristic approach for solving combinatorial optimization problems involving many variables. In recent years, simulated annealing has been employed in many diverse areas, such as very-large-scale integrated (VLSI) chip floorplanning, traveling salesman problem, image processing, and physical design of computers, to name a few. It is a probabilistic method based on ideas from statistical mechanics (Kirkpatrick et al., 1983), which deals with the behavior of systems having many degrees of freedom in thermal equilibrium at finite temperatures. Systems involving liquid metals freeze and crystallize or cool and anneal. At high temperatures, the molecules of liquid metals exhibit greater thermal mobility. If such a system is cooled slowly (i.e., *annealed*), the atoms orient themselves to form a pure crystal, thus attaining the lowest energy state of the system. On the other hand, if the liquid metal is cooled quickly (i.e., *quenched*), it does not reach this minimum energy state, but rather attains a polycrystalline or amorphous state possessing high energy.

The behavior of atoms in the presence of a heat bath is governed by the temperature. At each temperature, the system is allowed to attain thermal equilibrium. The probability (Pr) of such a system of being in a state with energy E is given by the Boltzmann distribution:

$$Pr(E) = \frac{1}{Z_t}\exp\left(\frac{-E}{K_b T}\right), \tag{11}$$

where K_b is the Boltzmann's constant (1.3806×10^{-23} J/K) and $1/Z_t$ is a normalization factor (Collins et al., 1988). In simulated annealing, the objective function (usually cost), is analogous to the energy of the system. The aim of such a problem then is to minimize the cost/energy denoted by $f(\bar{X})$, where $\bar{X} = (X_1, X_2, \ldots, X_N)$ represents a particular configuration of the system. To observe the behavior of the system, the system is perturbed from its present state to another state. These individual perturbations are referred to as neighborhood moves. It must be noted, however, that simulated annealing requires that these moves be Markov chains, since it is only in such cases that simulated annealing is guaranteed to attain global optimum, asymptotically (van Laarhoven and Aarts, 1987).

The behavior of a system subject to such a neighborhood move is determined from an observation of the objective function. If the configuration results in a lower energy state, the move is accepted. However, if the move results in a higher energy state, the move is still accepted according to Metropolis criteria [accepted with a probability $\exp(-\Delta E/K_b T)$]. Thus, a large percentage of uphill moves are accepted at high temperatures. The system is allowed to reach thermal equilibrium at each temperature, which is then lowered, and the annealing process continues until the system reaches a certain "freezing" temperature.

In mixed-discrete optimization, both discrete and continuous variables are involved; hence, such problems are combinatorial in nature. Such a solution space is much more complicated than that of a nonlinear optimization problem involving continuous variables. These problems are characterized as having multiple local minima in the solution space. Most existing algorithms on nonlinear optimization are developed to enhance the efficiency of the search procedure. For nonconvex objective function or constraints, these algorithms are unable to overcome local optima. However, since simulated annealing offers the alternative of a random uphill move, the system can "jump out" of a local minimum toward a global minimum. Simulated annealing thus combines standard iterative improvement and random uphill jumps to ensure that the system is not confined to a local minimum. A treatise on the choice of control parameters essential for such an annealing schedule (commonly called cooling schedule) is detailed in van Laarhoven and Aarts (1987).

Neighborhood moves in simulated annealing

In simulated annealing, the changes in the configuration of the system is realized through a neighborhood move from one point to another. For continuous variables, a neighborhood move is analogous to "a random walk with a bias." Based on theoretical results, a formulation of moves for the different class of variables can be devised as (Zhang and Wang, 1993):

- For integer variables, a move is defined as a change of a

variable from an integer value to another with a random step size.

- For zero-one variables, a move is defined as a change of a variable from zero to one or *vice versa:*

$$x_i' = \begin{cases} 1, & \text{if } x_i = 0 \\ 0, & \text{if } x_i = 1. \end{cases}$$

- For discrete variables, a move is defined as a change of a variable from one discrete value to another.
- For continuous variables, a move is defined as a random change of one variable:

$$x_i' = x_i + [2 \times \text{rand}(0,1) - 1]C_s,$$

where
x_i = value of the decision variable prior to the move
x_i' = value of the decision variable after the move
C_s = step size of the continuous variable

For continuous variables, there are other move sequences, a detailed treatise of which is presented elsewhere (Vanderbilt and Louie, 1984; Bohavchevsky et al., 1986). Further, it must be realized that for continuous variables, the step size of a neighborhood move should be small when the temperature is near "freezing" to obtain accurate solutions. Further, in multivariable problems, the neighborhood move can be realized in two ways: (1) moving one variable at a time (orthogonal move), or (2) moving all variables simultaneously (combined). In our approach, both the strategies have been employed in order to arrive at the final solution. Based on observations, orthogonal moves are better for constrained optimization problems, while the combined move approach is better for unconstrained optimization problems.

Penalty Function Approach

In a previous section, the importance of the confidence interval in determining the sample size was presented. In this section, we introduce the idea of penalty, which takes into account the confidence interval in the synthesis under uncertainty of large-scale combinatorial optimization problems. This leads us to formulate the new variant of the simulated annealing algorithm, the stochastic annealing algorithm.

In stochastic annealing, the cooling schedule is used to decide the weight on the penalty term for imprecision in the probabilistic objective function (by "stochastic annealing" we refer to the annealing of a probability or stochastic function. It must be realized that the simulated annealing algorithm is a stochastic algorithm inherently, since the moves are determined probabilistically. For our purposes, however, we will refer to the annealing of a deterministic objective function simply as simulated annealing). The choice of a penalty term, on the other hand, must depend on the error bandwidth of the function that is optimized, and must incorporate the effect of the number of samples. For the expected value as an objective function, the penalty term depends on the error bandwidth $2s/\sqrt{m}$. In process synthesis, there are a few cases in which one is interested in finding the decision variables for which the uncertainty is minimum. This is realizable in one form by minimizing the standard deviation of the expected function. Estimates of the lower and upper bounds of the standard deviation are obtained from the chi-square (χ) distribution, which for a random sample from a normal distribution with mean μ and standard deviation σ is given by

$$\sqrt{\frac{(m-1)s^2}{\chi^2_{\alpha/2}}} \quad \text{and} \quad \sqrt{\frac{(m-1)s^2}{\chi^2_{1-(\alpha/2)}}},$$

respectively. The penalty term in this case is based on half the difference between the upper and lower bound of the standard deviation.

The new objective function in stochastic annealing, therefore, consists of a probabilistic objective value and the penalty function, which is represented for the mean (Eq. 12) and the standard deviation (Eq. 13) as follows:

$$E(\text{cost}) = \frac{\sum_{i=1}^{N_{\text{samp}}} z_i}{N_{\text{samp}}} + b(t)\frac{2s}{\sqrt{N_{\text{samp}}}} \tag{12}$$

and

$$s(\text{cost}) = \sqrt{\frac{\sum_{i=1}^{N_{\text{samp}}}(z_i - \bar{z})^2}{N_{\text{samp}} - 1}} + \frac{b(t)}{2}\left(\sqrt{\frac{(N_{\text{samp}}-1)s^2}{\chi^2_{1-(\alpha/2)}}} - \sqrt{\frac{(N_{\text{samp}}-1)s^2}{\chi^2_{\alpha/2}}}\right). \tag{13}$$

In the preceding equations, the first term represents the real objective function and all other terms following the first term signify the penalty function.

The weighting function $b(t)$ can be expressed in terms of the temperature levels. At high temperatures, the sample size can be small, since the algorithm is exploring the functional topology or the configuration space to identify regions of optima. As the system gets cooler, the algorithm searches for the global optimum; consequently, it is necessary to take more samples to get more accurate and realistic objectives/costs. Thus, $b(t)$ increases as the temperature decreases. Based on this observation, an exponential function for $b(t)$ can be devised as

$$b(t) = \frac{b_o}{k^t}, \tag{14}$$

where b_o is small (e.g., 0.001), k is a constant that governs the rate of increase, and t is the current temperature level.

Stochastic Annealing Algorithm: A Variant of the Simulated Annealing Algorithm

The stochastic annealing algorithm minimizes the CPU time by balancing the trade-off between computational efficiency and solution accuracy by the introduction of a penalty term in the objective function. This is necessary, since at high temperature the algorithm is mainly exploring the solution space and does not require precise estimates of any probabilistic function. The algorithm must select a large number of

samples, as the solution is near the optimum. The weight of the penalty term, as mentioned before, is governed by $b(t)$, and is based on the annealing temperature. Based on these ideas, the stochastic annealing algorithm is as follows:

1. Initialize variables: T_{initial}, T_{freeze}, accept and reject limits, initial configuration S.
2. If $(T > T_{\text{freeze}})$ then perform the following loop (a–h) N (number of moves at a given temperature) times.
   (a) Generate a move S' from the current configuration S as follows:
      i. Select the number of samples, N_{samp} by a random move.

If $rand(0, 1) \leq 0.5$ then

$$N_{\text{samp}} = N_{\text{samp}} + 5 \times \text{rand}(0,1)$$

otherwise,

$$N_{\text{samp}} = N_{\text{samp}} - 5 \times \text{rand}(0,1)$$

      ii. Select the decision variables (zero-one, integer, discrete and continuous variables).
   (b) Generate N_{samp} samples of the uncertain parameters.
   (c) Perform the following loop [c(i)–c(ii)] N_{samp} times.
      i. Run the model.
      ii. calculate the objective function cost (S').
   (d) Evaluate the expected value $E[\text{cost}(S')]$ and $s[\text{cost}(S')]$ of the cost function.
   (e) Generate the weighting function $b(t) = b_o/k^t$
   (f) Calculate the modified objective function:

$$\text{Obj}(S') = E[\text{cost}(S')] + b(t)\frac{2s}{\sqrt{N_{\text{samp}}}}$$

or

$$\text{Obj}(S') = \sqrt{\frac{\sum_{i=1}^{N_{\text{samp}}}\{\text{cost}(S') - E[\text{cost}(S')]\}^2}{N_{\text{samp}}}} + \frac{b(t)}{2}\left(\sqrt{\frac{(N_{\text{samp}}-1)s^2}{\chi^2_{1-(\alpha/2)}}} - \sqrt{\frac{(N_{\text{samp}}-1)s^2}{\chi^2_{\alpha/2}}}\right)$$

      vii. Let $\Delta = \text{Obj}(S') - \text{Obj}(S)$
      viii. For a minimization problem, if $\Delta \leq 0$, then accept the move. Set $S = S'$; else if $(\Delta \geq 0)$, then accept with a probability $\exp(-\Delta/T)$.
3. If $T > T_{\text{freeze}}$, set $T = \alpha T$ and return to 2.
4. Stop.

Example Problem

The salient features of the stochastic annealing algorithm are best understood through a simple example involving integer and continuous variables. Consider a simple function:

$$\text{cost} = (u_1 y_1 - 3)^2 + (u_2 y_2 - 3)^2 + 2.0(x_1^2 - x_2)^2 + (x_1 - 1)^2, \quad (15)$$

where

u_1, u_2 = uncertain parameters
y_1, y_2 = integer variables
$x_1\ x_2$ = continuous variables.

The uncertain parameter u_1 was obtained from a uniform distribution [uniform (0.9, 1.1)] while u_2 was obtained from a normal distribution [normal (0.8, 1.2)] using Latin hypercube sampling. The bounds for the integer and continuous variables are as follows:

$$1 \leq y_1 \leq 4$$
$$1 \leq y_2 \leq 5$$
$$0 \leq x_1 \leq 6$$
$$0 \leq x_2 \leq 5.$$

The problem can be stated as

$$\text{Minimize } E(\text{cost}) \text{ or Minimize } s(\text{cost}),$$

subject to

$$1 \leq y_1 \leq 4$$
$$1 \leq y_2 \leq 5$$
$$0 \leq x_1 \leq 6$$
$$0 \leq x_2 \leq 5.$$

From observations, the minimum of the cost function is zero and occurs when $y_1 = y_2$ approximately equals 3, $x_1 = x_2 = 1$, and the average value of the uncertain parameters u_1 and u_2 are set to unity.

In Figure 5, the objective function [in this case E(cost)] is plotted against the annealing temperature. The objective function, E (cost), after every "move" was computed subject to the given uncertainties, using 100 samples and no penalty was imposed on the number of samples. On the other hand, Figure 6 shows the results of the run using the stochastic annealing algorithm, which incorporates the penalty function approach. The optimum reached in both cases was essentially the same, but the stochastic annealing algorithm took on an average 32 samples per temperature level to arrive at the minimum. A similar result was observed when we considered the problem of minimizing the standard deviation. Figure 7 shows the objective function, s(cost), vs. the annealing temperature for 1,000 samples using the simulated annealing algorithm. In contrast, the stochastic annealing algorithm selected on an average 89 samples per temperature level to arrive at the same minimum (Figure 8). The results are summarized in Table 1.

It is easily observed that the stochastic annealing algorithm uses less CPU time and finds the optimum using a lower number of samples. It is envisaged that the savings in the CPU time can be really significant in the analysis of large flowsheets. The stochastic annealing algorithm can enhance the design capability in process synthesis problems, as they can save computational time without sacrificing the precision of the optimal solution.

The stochastic annealing algorithm seeks to minimize a given objective subject to the uncertainties by allowing the trade-off between accuracy and computational efficiency. This

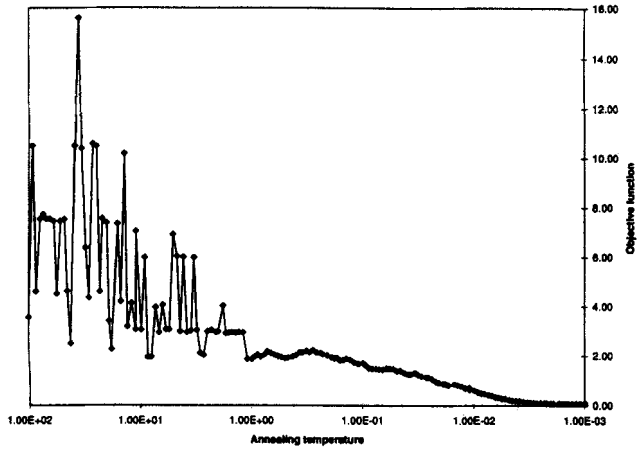

Figure 5. Objective function, *E*(cost) vs. annealing temperature for fixed number (100) of samples.

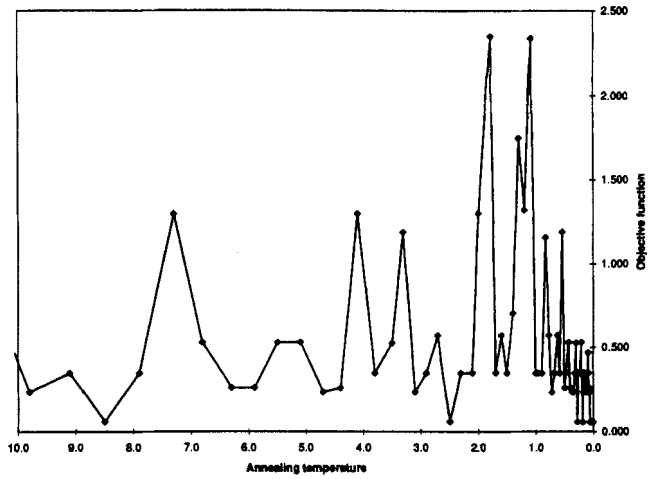

Figure 7. Objective function, *s*(cost) vs. annealing temperature for fixed number (1,000) of samples.

trade-off between accuracy and computational efficiency. This trade-off is better observed from Figure 9, which shows the number of samples and the penalty function [as a percentage of the objective function, in this case E(cost)] against the annealing temperature. Figure 10 shows the same for the standard deviation. Initially, the temperature T is high, and the algorithm is exploring the configuration space or the functional topology. The term, $b(t) = b_o/k^t$ in the penalty term is small, as t is small; but since the annealing begins with only a few samples, the penalty is high. This can increase the overall objective function. Thus, although it may seem that uphill moves are discouraged, the high temperature in the initial phase of annealing allows few uphill moves. Consequently, if uphill moves are not undertaken, the number of samples does not change and the probability function is evaluated using few samples. As the system anneals, T decreases, and as a consequence, $b(t)$ increases. It is observed that the effect of the weighted term, $b(t)$, increases the effect of the penalty function on the overall objective function. At low temperatures, the moves are mostly downhill; as a result, the algorithm tends to accept more samples to compute the objective function accurately. Finally, when the temperature is close to "freezing," all the moves are essentially downhill, and the weighted term does not allow the number of samples to increase indefinitely. This trade-off between accuracy and efficiency is the significant aspect of the algorithm, and allows the selection of an optimum number of samples, without loss of precision in the computation of the objective function.

HDA Process

The stochastic annealing algorithm has been used to synthesize a benchmark process in chemical engineering—the HDA (hydrodealkylation) process. For this purpose, a synthesis capability has been built around the public version of the ASPEN chemical process simulator. The details of this implementation in ASPEN are elaborated elsewhere (Chaudhuri and Diwekar, 1995). The HDA Process has been extensively studied by Douglas, starting from a hierarchical approach to the design problem (Figure 11). The problem we propose to solve is the selection of the flowsheet and some of the operating conditions that maximize profit. Also, we will assume that some of the cost parameters are uncertain and are represented as probability distributions. In order to illus-

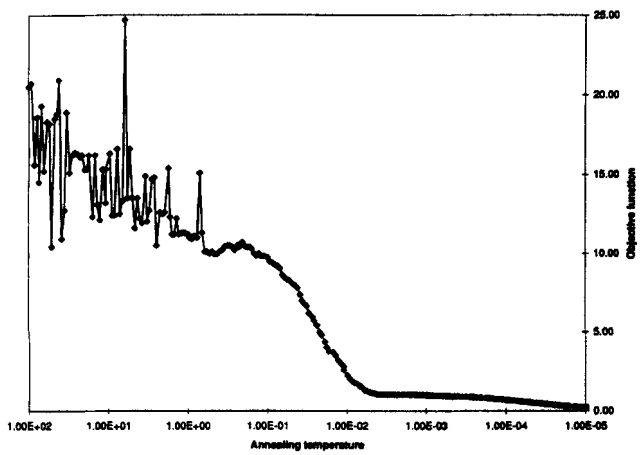

Figure 6. Objective function, *E*(cost) vs. annealing temperature for stochastic annealing which selects the number of samples.

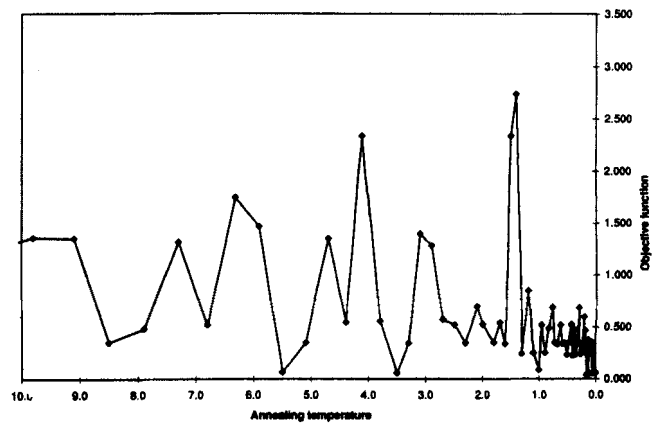

Figure 8. Objective function, *s*(cost) vs. annealing temperature for stochastic annealing which selects the number of samples.

Table 1. Results for the Example Problem*

Function	Algorithm	Samp./ Temp. Level	$y1$	$y2$	$x1$	$x2$	Optimum	CPU Time (s)
Exp. val.	Sim. Ann.	100	3	3	1.0001	1.0002	0.067	93.26
	Stoch. Ann.	32	3	3	1.0007	1.0008	0.069	74.54
Std. Dev.	Sim. Ann.	1,000	3	3	1.07021	0.93646	0.058	114,360.7
	Stoch. Ann.	89	3	3	1.07627	0.93536	0.059	61,841.8

*The CPU time does not include the sampling time. However, the total sampling time observed was much less compared to the execution time of the algorithms. In reality, the sampling time is usually a small percentage of the CPU time charged in the synthesis of large flowsheets. All computations were performed on VAX-4000.

trate the computational efficiency achieved by using the stochastic annealing algorithm, which automatically selects the optimum number of samples, the process flowsheet was also run using simulated annealing with fixed sample size.

Process description

The pertinent reactions of the HDA process are

$$\text{toluene} + \text{hydrogen} = \text{benzene} + \text{methane}.$$

In addition to this desired reaction, an undesired reaction:

$$\text{benzene} = \text{diphenyl} + \text{hydrogen}$$

also occurs. These homogeneous gas-phase reactions occur in the range of 894 K and 974 K. At lower temperatures, the reaction of toluene to produce benzene is too slow, while at high temperature substantial hydrocracking occurs. The pressure is maintained at 3.45 MPa in the reactor. Also, a molar ratio of at least 5:1 hydrogen to aromatics is maintained to prevent coking, and the reactor effluents must be quenched to 894 K to prevent coking in the heat-exchanger following the reactor.

The raw hydrogen stream has a purity of 95% (the rest is methane) and is mixed with a fresh inlet stream of toluene, recycled hydrogen, and toluene streams. These feed streams must be heated before being fed to the reactor. The reaction is exothermic and can be carried out in either an adiabatic reactor ($y1 = 1$) or an isothermal reactor ($y2 = 1$). The reactor effluent contains unreacted hydrogen, toluene, product benzene, and the undesired product methane and diphenyl, which is then quenched. The quenched stream is further cooled to condense the aromatics and separate the noncondensables (methane and hydrogen) in a flash separator.

To prevent the buildup of methane in the system, the vapor stream from the flash separator is purged; the remainder containing valuable hydrogen is recycled. A portion of the liquid stream from the flash separator is used to quench the reactor product stream, and the rest is sent to the liquid separation system. Since not all of the hydrogen and methane can be separated from the aromatics in the flash unit, most of it is removed in a distillation (or stabilizing) column ($y3 = 1$) or a flash separator ($y4 = 1$) operating at a lower pressure than the previous flash unit. The liquid stream contains essentially benzene, toluene, and diphenyl, which are separated in a sequence of distillation columns. The flash column separates benzene from toluene and diphenyl, while the second column separates toluene from undesired diphenyl.

The objective function is to maximize the annualized profit. The cost model is represented by linear fixed-charge costs and the data are given in Table 2, which also shows the uncertainties associated with some of the cost factors.

The mixed-discrete optimization problem contains four binary (0-1) variables and six continuous variables: furnace temperature, temperature of the isothermal reactor, conversion in the reactors, molar flow rate of the hydrogen and toluene streams, and seven constraints. The optimal design configuration is shown in Figure 12. The results (Table 3 indicate that stochastic annealing is able to attain the same optimal design configuration obtained using simulated annealing with a fixed sample size, but by stochastic annealing algorithm achieved 80% savings in CPU time and consequently

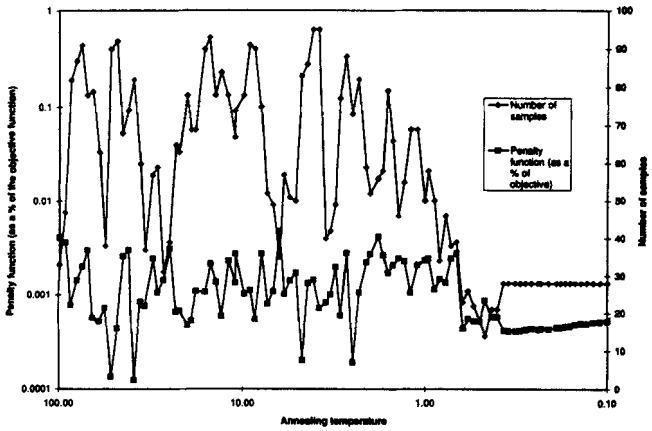

Figure 9. Effect of the number of samples on the penalty term for the expected value, *E*(cost) for stochastic annealing.

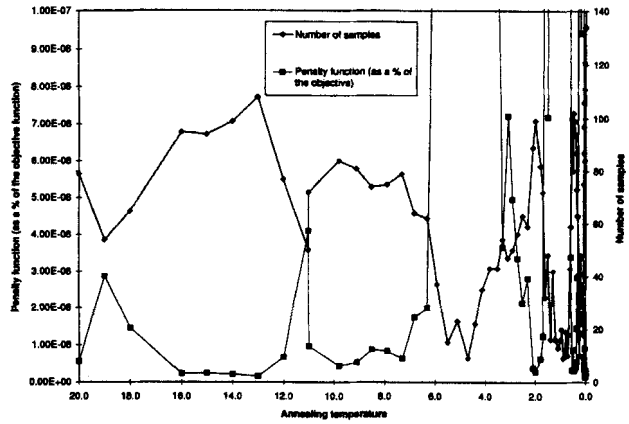

Figure 10. Effect of the number of samples on the penalty term for the standard deviation, *s*(cost) for stochastic annealing.

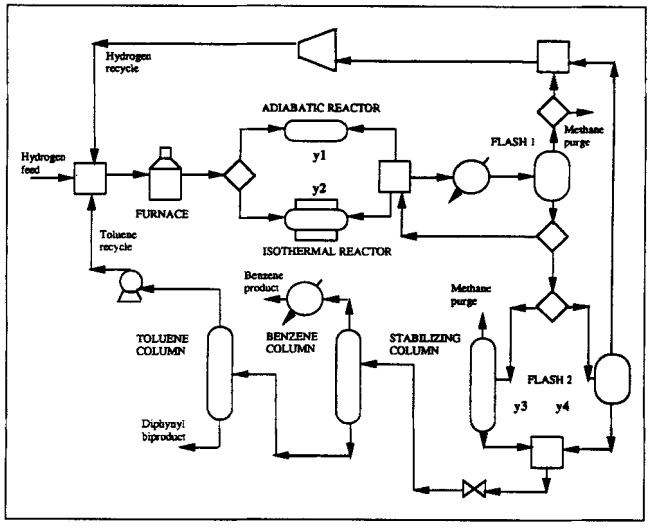

Figure 11. Superstructure of the HDA Process (Douglas, 1988).

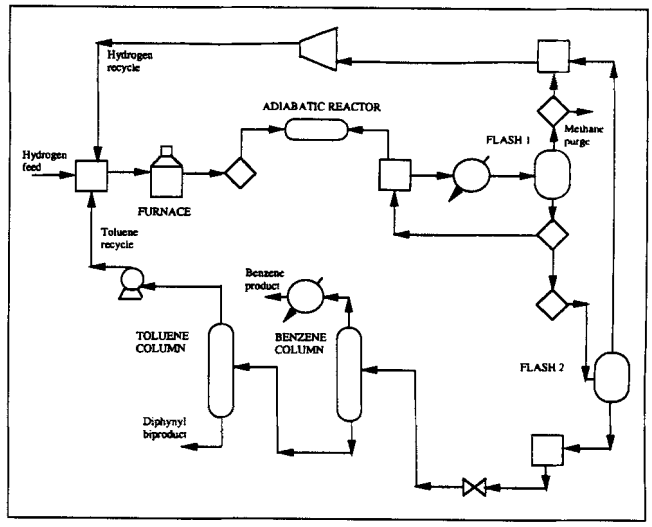

Figure 12. Optimal design configuration for the HDA Process.

provides an automated, efficient approach to stochastic synthesis.

Conclusion

The stochastic annealing algorithm is designed to optimize an objective function, which is an estimate of any statistical parameter, such as the expected value or the standard deviation of any output parameter. We have modified the standard simulated annealing algorithm to choose not only the decision variables, both discrete and continuous, but also the number of samples. The number of samples is less when the temperature is high and the algorithm is exploring the entire configuration space. When the system is close to "freezing," the algorithm allows for an increased number of samples, but at the same time does not allow the selection of "too many" samples. This is actually incorporated in the algorithm as a penalty term, which uses the confidence interval or the bounds for the objective function to enable the selection of the number of samples, and the penalty term is weighted, based on the annealing temperature. The stochastic annealing algorithm has been applied to the synthesis of a benchmark chemical process—the hydrodealkylation of toluene to produce benzene, which is frequently referred to in the literature. Both simulated annealing with fixed sample size and stochastic annealing were used to analyze the flowsheet. The stochastic annealing has been shown to find the optimal flowsheet configuration, with 80% savings in CPU time. This new, innovative synthesis tool therefore holds great promise in the synthesis of large-scale complex chemical processes under uncertainty.

Acknowledgment

This work has been funded by a DOE/METC Research Grant. The authors would like to thank Laura Painton for her helpful suggestions.

Table 2. Cost Data for HDA Problem

Feedstock or pdt/by-pdt.	Cost	Uncertainty	
Hydrogen (95% H_2, 5% CH_4)	\$2.50/kmol	normal (0.6, 1.4)	
Toluene (100% toluene)	\$14.0/kmol	triangular (0.6, 1.4)	
Benzene ($\geq$ 90%)	\$19.90/kmol	normal (0.7, 1.3)	
Diphenyl	\$11.84/kmol	lognormal (0.6, 1.1)	
Hydrogen purge (heat val.)	\$1.08/kmol		
Methane purge (heat val.)	\$3.37/kmol		
Utilities	**Cost**		
Electricity	\$0.04/kWh		
Heating (steam)	\$8.0/10^6 kJ		
Cooling (water)	\$0.7/10^6 kJ		
Fuel	\$4.0/10^6 kJ		
Investment costs, \$$10^3$ yr^{-1}	**Fixed-charge costs**	**Uncertainty**	**Lin. coeff.**
Compressor	\$7.155	normal (0.8, 1.2)	0.815 × bhp (kW)
Stabilizing col.	\$1.126	uniform (0.6, 1.4)	0.375 × No. of trays
Benzene col.	\$16.3	—	1.55 × No. of trays
Toluene col.	\$3.90	—	1.12 × No. of trays
Furnace	\$6.20	—	1,172 × duty (10^6 kJ/yr)
Adiab. reactor	\$74.3	—	1.257 × vol (m^3)
Iso. reactor	\$92.875	—	1.571 × vol (m^3)

Table 3. Results for the HDA Problem

Decision Variables	Sim. Annealing (Sample Size Fixed)	Stoch. Annealing
$y1$	1	1
$y2$	0	0
$y3$	0	0
$y4$	1	1
Conversion	0.603	0.609
Reactor temp.	863.5 K	866.9 K
Furnace temp.	891.9 K	893 K
Molar flow rate (hydrogen feed)	252.44 kmol/h	252.62 kmol/h
Molar flow rate (toluene feed)	126.84 kmol/h	127.31 kmol/h
CPU time	74,000 s	14,516 s
Maximized Profit, $\$/yr \times 10^3$	621.1	649.7

Notation

E = energy of a state
$E(Z)$ = expected value of the objective function
y_i = a particular observation
Z_t = reciprocal of the normalization factor

Literature Cited

Bohachevsky, I. O., M. E. Johnson, and M. L. Stein, "Generalized Simulated Annealing for Function Optimization," *Technometrics*, **28**, 209 (1986).

Chaudhuri, P. D., and U. M. Diwekar, "Synthesis under Uncertainty with Simulators," *Comput. Chem. Eng.*, under review (1995).

Ciric, A. R., and S. Huchette, "Multiobjective Optimization and Nonlocal Sensitivity in Process Source Reduction Problems," AIChE Meeting, Houston (1993).

Collins, N. E., R. W. Eglese, and B. L. Golden, "Simulated Annealing—An Annotated Bibliography," *Am. J. Math. Mgmt. Sci.*, **8**, 209 (1988).

Das, H., P. T. Cummings, and M. D. LeVan, "Scheduling of Serial Multiproduct Batch Processes via Simulated Annealing," *Comput. Chem. Eng.*, **14**, 1351 (1990).

Diwekar, U. M., and E. S. Rubin, "Stochastic Modeling of Chemical Processes," *Comput. Chem. Eng.*, **15**, 105 (1991).

Diwekar, U. M., I. E. Grossmann, and E. S. Rubin, "An MINLP Process Synthesizer for a Sequential Modular Simulator," *Ind. Eng. Chem. Res.*, **31**, 313 (1992a).

Diwekar, U. M., H. C. Frey, and E. S. Rubin, "Synthesizing Optimal Flowsheets: Applications to IGCC System Environmental Control," *Ind. Eng. Chem. Res.*, **31**, 1927 (1992b).

Diwekar, U. M., "A Process Analysis Approach to Pollution Prevention," AIChE Symp. Ser. on Pollution Prevention through Process and Product Modifications, **90**, 168 (1994).

Dolan, W. B., P. T. Cummings, and M. D. LeVan, "Process Optimization via Simulated Annealing: Application to Network Design," *AIChE J.*, **35**, 725 (1989).

Douglas, J. M., "A Heirarchial Decision Procedure for Process Synthesis," *AIChE J.*, **31**(3), 353 (1985).

Douglas, J. M., *Conceptual Design of Chemical Processes*, McGraw-Hill, New York (1988).

Grossmann, I. E., "Mixed-Integer Programming Approach for the Synthesis of Integrated Process Flowsheets," *Comput. Chem. Eng.*, **9**(5), 463 (1985).

Grossmann, I. E., "Mixed-Integer Nonlinear Programming Techniques for the Synthesis of Engineering Systems," *Res. Eng. Des.*, **1**, 205 (1990).

Iman, R. L., and M. J. Shortencarier, "A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for use with Computer Models," NUREG/CR-3624, SAND83-2365, Sandia National Laboratory, Albuquerque, NM (1984).

Kirkpatrick, S., C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, **220**, 670 (1983).

Kocis, G. R., and I. E. Grossmann, "A Modeling and Decomposition Strategy for MINLP Optimization of Process Flowsheets," *Comput. Chem. Eng.*, **13**, 797 (1989).

Kravanja, Z., and I. E. Grossmann, "PROSYN—An MINLP Synthesizer," *Comput. Chem. Eng.*, **14**, 1363 (1990).

Linnhoff, B., in *Foundations of Computer-Aided Chemical Process Design*, Vol. II, R. S. H. Mah and W. D. Seider, eds., Engineering Foundation, New York, p. 537 (1981).

Milton, J. S., and J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and Computing Sciences*, 2nd ed., McGraw Hill, New York (1990).

Morgan, G. M., and M. Henrion, *Uncertainty—A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge Univ. Press, Cambridge, England (1990).

Nishida, N., G. Stephanopoulos, and A. Westerberg, "A Review of Process Synthesis," *AIChE J.*, **27**, 321 (1981).

Painton, L. A., and U. M. Diwekar, "Synthesizing Optimal Design Configurations for a Brayton Cycle Power Plant," *Comput. Chem. Eng.*, **18**, 369 (1994).

Painton, L. A., and U. M. Diwekar, "Stochastic Annealing under Uncertainty," *Eur. J. Oper. Res.*, **83**, 489 (1995).

Paulas, G. E., IV, and C. A. Floudas, "APROS—Algorithmic Development Methodology for Discrete-Continuous Optimization Problems," *Oper. Res. J.*, **37**, 902 (1989).

Pistikopoulos, E. N., and I. E. Grossmann, "Stochastic Optimization of Flexibility in Retrofit Design of Linear Systems," *Comput. Chem. Eng.*, **12**, 1215 (1988).

Rutenbar, R. A., "Simulated Annealing Algorithms: An Overview," *IEEE Circuits Devices Mag.*, **100**, 19 (1989).

Straub, D. A., and I. E. Grossmann, "Integrated Stochastic Metric of Flexibility for Systems with Discrete State and Parameter Uncertainties," *Comput. Chem. Eng.*, **14**, 967 (1990).

Vanderbilt, D., and S. G. Louie, "A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables," *J. Comput. Phys.*, **56**, 259 (1984).

van Laarhoven, P. J. M., and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, The Netherlands (1987).

Varvarezos, D., I. E. Grossmann, and L. T. Biegler, "Reduced SQP Strategies for Multiperiod Process Design Problems," AIChE Meeting, Houston (1993).

Zhang, C., and H. Wang, "Mixed-Discrete Nonlinear Optimization with Simulated Annealing," *Eng. Opt.*, **21**, 277 (1993).

Manuscript received Jan. 5, 1995, and revision received Apr. 24, 1995.