IIE Transactions (2002) **34**, 761–777

Hammersley stochastic annealing: efficiency improvement for combinatorial optimization under uncertainty

KI-JOO KIM and URMILA M. DIWEKAR*

CUSTOM (Center for Uncertain Systems: Tools for Optimization and Management), Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
E-mail: urmila@cmu.edu

Received October 2000 and accepted December 2001

This paper presents hierarchical improvements to combinatorial stochastic annealing algorithms using a new and efficient sampling technique. The Hammersley Sequence Sampling (HSS) technique is used for updating discrete combinations, reducing the Markov chain length, determining the number of samples automatically, and embedding better confidence intervals of the samples. The improved algorithm, Hammersley stochastic annealing, can significantly improve computational efficiency over traditional stochastic programming methods. This new method can be a useful tool for large-scale combinatorial stochastic programming problems. A real-world case study involving solvent selection under uncertainty illustrates the usefulness of this new algorithm.

1. Introduction

Optimization under uncertainty refers to that branch of optimization problems where there are uncertainties involved in the data or model, popularly known as stochastic programming problems. Stochastic programming gives the ability to optimize in the face of uncertainties and requires that the objective function and constraints be expressed in terms of some probabilistic representation (e.g., expected value, variance, fractiles, or most likely values). The general way to treat the probabilistic functional of the objective function and constraints is to use stochastic models instead of deterministic models in the problem formulation. Thus a stochastic optimization problem, where there are decision variables and uncertain parameters, can be viewed as:

$$\begin{aligned} (\mathbb{P}1) \quad & \min \; z = P_1[f(\mathbf{x}, \boldsymbol{\xi})], \\ \text{subject to} \quad & P_2[h(\mathbf{x}, \boldsymbol{\xi})] = 0, \\ & P_3[g(\mathbf{x}, \boldsymbol{\xi})] \leq 0, \\ & \boldsymbol{x} \in X, \; \boldsymbol{\xi} \in \Xi, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a vector of decision variables, and ξ is a vector of uncertain parameters of the domain Ξ. The performance metric to be optimized is represented by a probabilistic function P_1, and the model equality and inequality constraints are defined by a set of probability functions P_2 and P_3, respectively. If P_i is the expected value, the above optimization problem becomes:

$$(\mathbb{P}2) \quad \min z = E_{\xi} f(\mathbf{x}, \boldsymbol{\xi}), \tag{2}$$

where E_{ξ} is the mathematical expectation with respect to ξ. The optimal solution and optimal value of problem ($\mathbb{P}2$) are x^* and z^*, respectively. The main difficulty of stochastic programming stems from evaluating the uncertain functions and their expectations. A generalized method to propagate the uncertainties is to use a sampling method. A common method is to propagate N_{samp} samples generated from the random values of ξ and optimize the following approximated problem:

$$(\mathbb{P}3) \quad \min \; z = \frac{1}{N_{\text{samp}}} \sum_{j=1}^{N_{\text{samp}}} f(\mathbf{x}, \boldsymbol{\xi}^j). \tag{3}$$

Similarly, the optimal solution and optimal value of this approximation problem are $\hat{x}$ and $\hat{z}$, respectively.

To solve this approximated stochastic approximation problem, the optimization and sampling technique for ξ are performed simultaneously. Thus the generalized stochastic framework for solving optimization under uncertainty problems involve two recursive loops: (i) the inner sampling loop; and (ii) the outer optimization loop. For example, sampling is embedded into the L-shaped method, which is an approximation of the nonlinear term in the objective function of stochastic programming problems (Dantzig and Glynn, 1990; Hingle and Sen, 1991).

* Corresponding author

In the sampling loop, MCS (Monte Carlo Sampling), a *pseudo*-random number generator, has been commonly used for representing ξ. Even though MCS generates independent and random samples of a given probability distribution, it is known that MCS requires a large number of samples, N_{samp}, to approximate the "true" mean or variance. Decreasing distance between the true and approximated optimal solutions $|x^* - \hat{x}|$, therefore, increases N_{samp} and results in a high computational intensity. To reduce the computational burden of stochastic programming problems, one can use either decomposition methods with sampling (Dantzig and Glynn, 1990; Higle and Sen, 1991; Shapiro and Homem-De-Mello, 2000) or efficient sampling methods. However, most of the decomposition methods require convexity conditions and dual-angular structures, and are only applicable to problems involving continuous decisions. The area of discrete optimization problems, the focus of this paper, contains computationally intensive stochastic programming problems.

Recently, a new *quasi*-random sampling technique referred to as Hammersley Sequence Sampling (HSS) has been proposed (Hammersley, 1960; Kalagnanam and Diwekar, 1997), that it has been shown to exhibit a better *homogeneity* over the multivariate parameter space. Further, for this new sampling technique, it is found that the number of samples required to converge to the different performance measures (such as mean, variance, or fractiles) of an output random variable, subject to input uncertainties, is lower than the crude MCS or the variance reduction techniques such as Latin Hypercube Sampling (LHS) (Iman and Conover, 1982). This rapid convergence property of HSS has important implications for stochastic programming, suggesting that precise estimates of any probabilistic function evaluations are achievable by taking a smaller sample size. These uniformity and faster convergence properties of HSS can be used for the outer optimization loop as well as the inner loop to achieve better computational efficiency with the stochastic programming problems involving discrete decision and continuous decision variables.

Simulated Annealing (SA), a probabilistic method based on ideas from statistical mechanics was developed by Kirkpatrick *et al.* (1983), and is applied to combinatorial optimization problems. In recent years, a new variant of SA called STochastic Annealing (STA) (Painton and Diwekar, 1995; Chaudhuri and Diwekar, 1996, 1999) was designed to efficiently optimize a probabilistic objective function ($\mathbb{P}4$) by automatically selecting the number of samples needed to approximate the uncertain surface. By adding a penalty term to ($\mathbb{P}3$), the STA can trade-off between solution efficiency and solution accuracy.

$$(\mathbb{P}4) \quad \min\ z = \frac{1}{N_{\text{samp}}} \sum_{j=1}^{N_{\text{samp}}} f(\mathbf{x}, \xi^j) + (\text{penalty term}). \quad (4)$$

where the penalty term is described in Section 3.2.

This paper focuses on improving the computational efficiency of the SA-based algorithms. The probabilistic nature of these SA-based algorithms is the basis of the hierarchical improvement presented in this paper. Hierarchical improvements of stochastic annealing for reducing computational intensity are achieved by using a new efficient sampling technique, HSS, both in the inner sampling and in the outer optimization loops. New improved SA-based stochastic programming algorithms are summarized in Table 1. In SA and its variants, each configuration that is a set of discrete decision variables is randomly generated from the previous configuration, and this update is generally based on a random probability given by a uniform distribution of the neighboring configurations. Because it is known that this probability can significantly affect the overall efficiency of annealing process (Van Laarhoven and Aarts, 1987), ESA (Extended Simulated Annealing) aims to improve this generation mechanism by using the *uniformity* property of HSS. The ESTA (Extended STochastic Annealing) algorithm not only improves the generation mechanism but also reduces the number of samples required in the inner sampling loop by using the *faster convergence* property of HSS. Finally, HSTA (Hammersley STochastic Annealing) applies an accurate confidence interval of the samples to the ESTA algorithm.

A real-world case study involving a solvent selection problem commonly encountered in pharmaceutical and chemical industries is presented to illustrate the usefulness of this HSTA algorithm. In this study, we have applied HSTA to a solvent selection problem under uncertainty whose aim is to find the best promising solvents with desired properties. All possible solvent molecules are generated by constructing their building blocks or groups, and their chemical and physical properties are estimated. HSTA is used to select groups to form molecules, impose uncertainties on property estimation methods, and determine whether or not the probabilistic objective functional is optimum.

This paper is structured as follows. Section 2 describes the generalized stochastic models involved in stochastic programming problems. Section 3 explains and demonstrates hierarchical improvements in the SA-based algorithms. Solvent selection under uncertainty is presented as a case study in Section 4. The last section concludes the paper.

Table 1. Three-level efficiency improvement in stochastic optimization

Level	*Algorithm*	*Improvement targets*
I	ESA	Optimizer (SA)
II	ESTA	Optimizer (SA) + Sampling technique
III	HSTA	Optimizer (SA) + Sampling technique + Confidence interval

2. Uncertainty propagation and sampling

This probabilistic or stochastic modeling procedure involves: (i) specifying the uncertainties in key input parameters in terms of probability distributions; (ii) sampling the distribution of the specified parameter in an iterative fashion; and (iii) propagating the effects of uncertainties through the process flowsheets and applying statistical techniques to analyze the results.

2.1. *Specifying uncertainty using probability distributions*

To accommodate the diverse nature of uncertainty, different distributions can be used for the uncertain parameters ξ. Some of the representative distributions are uniform, triangular, normal and lognormal distributions. The type of distribution chosen for an uncertain variable reflects the amount of information that is available. For example, uniform and log-uniform distributions represent an equal likelihood of a value lying anywhere within a specified range, on either a linear or logarithmic scale, respectively. The normal (Gaussian) distribution reflects a symmetric but varying probability of a parameter value being above or below the mean value. In contrast, the lognormal and some triangular distributions are skewed such that there is a higher probability of values lying on one side of the median than on the other. The beta distribution provides a wide range of shapes and is a very flexible means of representing variability over a fixed range. Finally, in some special cases, user-specified distributions can be used to represent any arbitrary characterization of uncertainty, including the fractile distribution (i.e., fixed probabilities of discrete values).

2.2. *Sampling techniques in stochastic modeling*

Once probability distributions are assigned to the uncertain parameters, the next step is to perform a sampling operation from the multi-variable uncertain parameter domain. Alternatively, one can use analytical methods to obtain the effect of uncertainties on the output. These methods, however, tend to be applicable to special kinds of uncertainty distributions and optimization surfaces only. The sampling approach provides wider applicability and is discussed below.

2.2.1. *Monte Carlo technique*

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

The main advantage of Monte Carlo methods lies in the fact that the results from any Monte Carlo simulation can be treated using classic statistical methods because of the randomness and independence of generated samples. Results can thus be presented in the form of histograms, and methods of statistical estimation and inference may be applied. Nevertheless, in most applications, the actual relationship between successive points in a sample has no physical significance. Hence, the randomness/independence for approximating a uniform distribution is not critical (Knuth, 1973). Moreover, the error of approximating a distribution by a finite number of samples depends on the distribution properties of the sample used for $U(0,1)$ rather than its randomness. Once it is apparent that the uniformity property is central to the design of sampling techniques, a constrained or stratified sampling technique such as Latin Hypercube Sampling (LHS) becomes appealing (Morgan and Henrion, 1990). LHS is one form of the variance reduction technique and is widely used in risk and decision analysis literature.

2.2.2. *Latin hypercube sampling*

Latin hypercube sampling (Iman and Conover, 1982) is one form of the stratified sampling technique which can yield more precise estimates of the distribution function. In LHS, the range of each uncertain parameter Ξ_i is subdivided into non-overlapping intervals of equal probability. One value from each interval is selected at random with respect to the probability distribution in the interval. The N_{samp} values thus obtained for Ξ_1 are paired in a random manner (i.e., equally likely combinations) with N_{samp} values of Ξ_2. These N_{samp} values are then combined with N_{samp} values of Ξ_3 to form N_{samp}-triplets, and so on, until N_{samp} k-tuplets are formed. The main drawback of this stratification scheme is that it is uniform in one-dimension, but does not provide the uniformity property in a k-dimensional hypercube.

2.2.3. *Importance sampling*

Stratified sampling techniques ensure that more samples are generated from high probability regions. On the other hand, importance sampling techniques guarantee full coverage of high consequence regions in the sample space, even if these regions are associated with low probabilities. This makes importance sampling techniques problem-dependent.

2.2.4. *Hammersley sequence sampling*

Recently, an efficient sampling technique, called Hammersley Sequence Sampling (HSS) and based on Hammersley points, has been developed by Kalagnanam and Diwekar (1997), which uses an optimal design scheme for placing N_{samp} points on a k-dimensional hypercube. This scheme ensures that the sample set is more representative of the population, showing better uniformity in the multi-dimensional uncertain surface, unlike Monte Carlo, Latin hypercube, and its variant, the Median Latin hypercube sampling techniques (Kalagnanam and Diwekar, 1997).

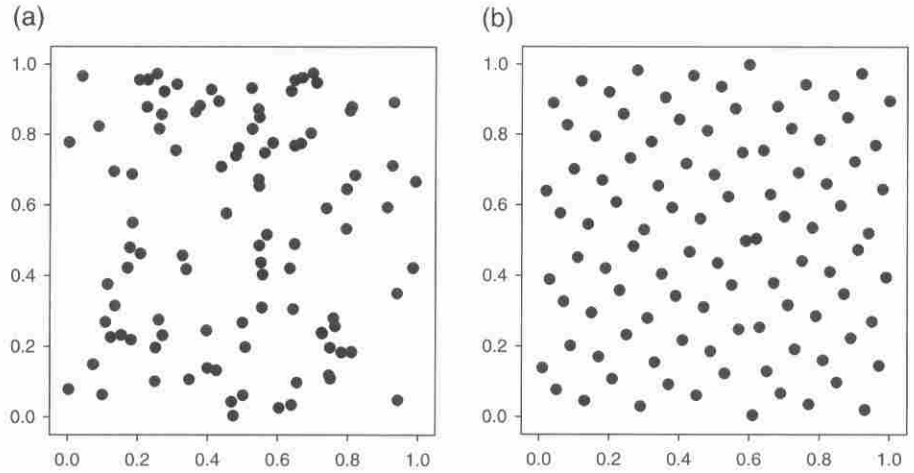

Fig. 1. Sample points (100) on a unit square using: (a) crude Monte Carlo sampling; and (b) the Hammersley sequence sampling technique.

Figure 1(a and b) shows samples generated by different sampling techniques on a unit square and provides a qualitative picture of how uniform the samples are. It is clear from Fig. 1(a and b) that HSS has a better uniformity property compared to the other sampling techniques. The main reason for this is that the Hammersley points that are one of the minimum discrepancy designs provide an optimal design for placing N_{samp} points on a k-dimensional hypercube. In contrast, stratified techniques such as the LHS technique are designed for uniformity along a single-dimension and then randomly paired for placement on a k-dimensional cube. Therefore, the likelihood of such schemes providing a good uniformity property on high-dimensional cubes is small.

The better uniformity property of HSS results in faster convergence to the "true" mean, variance, or fractiles of a function with multi-dimensional uncertainties. Since there are no analytic approaches (for stratified designs) to calculate the number of samples required for convergence to the "true" mean or variance, Kalagnanam and Diwekar (1997) conducted a large matrix of numerical tests and established that the HSS technique is at least three to 100 times faster than the MCS and LHS techniques and hence is a preferred sampling technique for uncertainty analysis as well as stochastic programming (Chaudhuri and Diwekar, 1999). This same uniformity property of the Hammersley sequence can be used to systematically improve the efficiency of the simulated annealing-nonlinear programming-based framework for optimization under uncertainty and is presented in the next section.

3. Hierarchical efficiency improvement

The hierarchical improvements to the SA-based algorithm for discrete optimization under uncertainty is described below. These improvements are at three levels: (i) the inner sampling loop; (ii) the outer discrete deterministic optimization loop; and (iii) the interaction between the optimization and the sampling loops.

3.1. *Extended Simulated Annealing (ESA)*

Simulated annealing is a probabilistic method for combinatorial optimization problems based on ideas from statistical mechanics. The analogy in SA is to the behavior of physical systems in the presence of a heat bath: in physical annealing, all atomic particles arrange themselves in a lattice formation that minimizes the amount of energy in the system, provided the initial temperature is sufficiently high and the cooling is carried out slowly. At each temperature T, the system is allowed to reach thermal equilibrium, which is characterized by the probability of being in a state with energy H given by the Boltzmann distribution function:

$$Pr(H) = \frac{1}{Z}\exp\left(-\frac{1}{k_B}\frac{H}{T}\right), \tag{5}$$

where $1/Z$ is a normalization factor and k_B is the Boltzmann's constant (1.3806×10^{-23} J/K). Furthermore, at each temperature level, the system should follow Markovian moves where the next move is only dependent on the current one, not the previous ones.

In SA, the objective function (usually cost) becomes the energy of the system, and the goal is to minimize the energy. Simulating the behavior of the system then becomes a question of generating a random perturbation that displaces a current "particle" (moving the system to another configuration). If the configuration S representing a set of discrete decision variables that results from the moves has a lower energy state, the move is accepted. However, if the move is to a higher energy state, the move is accepted according to the Metropolis criterion that is given by Van Laarhoven and Aarts (1987) to be:

$$\text{accepted if } Pr_{\text{accept}} \le A_{ij}$$
$$= \begin{cases} \exp\left(-\frac{\Delta H}{T}\right) & \text{if } \Delta H = H_j - H_i \ge 0, \\ 1 & \text{otherwise,} \end{cases} \quad \forall i, j \in S, \tag{6}$$

where A_{ij} is the acceptance probability for generating configuration j from i. The uphill moves can be accepted if a random probability (Pr_{accept}) is less than or equal to the A_{ij}. Hence, the Metropolis criterion implies that at high temperatures, a large percentage of uphill moves are accepted. However, as the temperature gets colder, only a small percentage of uphill moves are accepted. Note that these uphill moves are not allowed in conventional local optimization algorithms. After the system has evolved to thermal equilibrium at a given temperature, then the temperature is lowered, and the annealing process continues until the system reaches a certain "freezing" temperature determined *a priori*. Thus, SA combines both iterative improvements in local areas and random jumpings to help ensure that the system does not becomes stuck in a local optimum.

As SA is a probabilistic method, several random probability functions are involved in this procedure. A_{ij} represents the acceptance probability, and one or more generation probabilities G_{ij} are used to generate subsequent configurational moves. It is known that the efficiency of the annealing algorithm is affected little by the use of different probability distributions of Pr_{accept} (Van Laarhoven and Aarts, 1987). However, G_{ij} for generating configuration j from i at each temperature can significantly affect the overall efficiency of the annealing process. The cooling schedule is strongly dependent on G_{ij}: if the cooling is fast, then G_{ij} should cover a wider range of the configuration space at each temperature level. Generally G_{ij} is a random probability given by the uniform distribution within the neighborhood. Thus recent research efforts for SA improvement have been focused on modifying or changing G_{ij}. These new simulated annealing algorithms differ mainly in the choice of G_{ij} and the cooling schedule (Salazar and Toral, 1997). Among the proposed simulated annealing variants, Fast Simulated Annealing (FSA) (Szu and Hartley, 1987) and Hybrid Simulated Annealing (HSA) (Salazar and Toral, 1997) are worth mentioning. G_{ij} in FSA has a Gaussian-like peak and Lorentzian long tails which can make an occasional long jump from the current configuration to increase the speed of annealing. However this G_{ij} cannot guarantee uniform coverage of the moves over the configuration surface. HSA applies the Hybrid Monte Carlo method to obtain G_{ij} in which the design variable $\mathbf{x}$ and a Gaussian-like auxiliary momenta $\mathbf{p}$ are mapped using Hamilton's equations of motion. The acceptance probability is similar to the Metropolis criterion, but the energy difference ΔH is replaced by the Hamiltonian function difference $\Delta\mathscr{H}(\mathbf{x}, \mathbf{p})$. Although this algorithm is found to be very fast, HSA requires an evaluation of the derivative of the objective function, $-\partial f(\mathbf{x})/\partial x_i$, for mapping and hence this destroys one of the advantages of the standard simulated annealing algorithm (that SA does not require derivative information).

The G_{ij} generation of the current annealing algorithms rely on pseudo-random number generators like the crude MCS which can result in randomly clustered moves over the configuration surface as shown in Fig. 1(a). Therefore, more moves or generations are required to cover the whole configuration surface evenly, and this results in a longer *Markov chain length* (i.e., number of moves) at each temperature level. As described in the previous section, the HSS technique (Fig. 1(b)), a *quasi*-random number generator, can generate uniform samples over the k-dimensional hypercube. In this work, we use HSS to generate the G_{ij} for SA and derive a new SA algorithm called Extended Simulated Annealing (ESA). It should be noted in using the HSS for ESA, one has to keep the k-dimensional uniformity property of HSS by generating the k probabilities for G_{ij} from configuration i in a complete space of a Markov chain length. To illustrate this, consider the following Stochastic Integer Programming (SIP) problem from Birge and Louveaux (1997):

$$\begin{aligned} \min \quad & -2y_1 - 3y_2, \\ \text{subject to} \quad & y_1 + 2y_2 \le \xi_1 - x_1, \\ & y_1 \le \xi_2 - x_2, \\ & y \ge 0, \text{ integer,} \end{aligned} \tag{7}$$

where $\boldsymbol{\xi} = (2,2)^{\mathrm{T}}$ or $\boldsymbol{\xi} = (4,3)^{\mathrm{T}}$, each with an equal probability, and the current iterate point is at $\mathbf{x} = (0,1)^{\mathrm{T}}$. The optimal solution is at $\mathbf{y} = (2,1)^{\mathrm{T}}$. The implementation of SA to this example requires four probabilities: one for assigning $\boldsymbol{\xi}$, two for discrete up and down moves of $\mathbf{y}$, and one for the Metropolis criterion. In traditional SA algorithms, these probabilities are established by generating a *pseudo*-random number based on MCS at a time. However, to exploit the k-dimensional uniformity of HSS, these four probabilities are generated together by generating all the N (number of moves at each temperature, i.e., Markov chain length) quasi-random numbers based on HSS at once, and then using them one at a time.

To compare the performance of ESA with SA, both MCS and HSS ways are used to generate the four probabilities. The average value of the total moves, which are the number of moves at each temperature level (i.e., each step) multiplied by annealing steps, is obtained from 15 different initial conditions. The total moves for the SIP with the HSS technique is found to be 530 while the MCS technique required a total of 1331 moves, a significant saving.

To further demonstrate the efficiency improvement of the ESA over the SA, the following examples are tested:

$$\text{Example I:} \quad f(\mathbf{y}) = \sum_{i=1}^{ND} y_i^2, \tag{8}$$

$$\text{Example II:} \quad f(\mathbf{y}) = \sum_{i=1}^{y_1} \Big\{ (y_1 - 3)^2 + (y_2(i) - 3)^2 + (y_3(i) - 3)^2 \Big\}, \tag{9}$$

$$\text{Example III:} \quad f(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{ND} \left(x_i - \frac{i}{ND} \right)^2 + \sum_{i}^{ND} y_i^2 - \prod_{i}^{ND} \cos(4\pi y_i), \tag{10}$$

$$\text{Example IV:} \quad f(\mathbf{y}) = \left[0.002 + \sum_{j=1}^{25} \Big\{ j + (y_1 - a_j)^6 + (y_2 - b_j)^6 \Big\}^{-1} \right]^{-1}, \tag{11}$$

where **x** denotes a vector of continuous variables and **y** denotes a vector of discrete variables. Example I is a multi-dimensional parabolic function taken from Salazar and Toral (1997) where *ND* is the dimension of the function. This example has one global optimum at zero for all decision variables. The second example, a pure combinatorial problem, appears in Painton and Diwekar (1994) where the objective space is *discontinuous* with respect to y_1. This example also has one global optimum when y_1 is three and all $y_2(i), y_3(i)$ are three ($i = 1, \ldots, y_1$). Since the third example involves discrete and continuous decision variables, this example function is a MINLP problem. As an alternative to MINLP, a coupling of SA and NLP, SA-NLP (Narayan *et al.*, 1996), is used to solve this problem. This function has one global optimum ($f(\mathbf{x},\mathbf{y}) = -1$) and many local optima. The last example is De Jong's test function (De Jong, 1981), which is commonly used for benchmarking SA algorithms and genetic algorithms. The constants, a_j and b_j, have the following 25 components:

$$a_j = \{-32, -16, 0, 16, 32, -32, -16, 0, 16, 32, \ldots, -32, -16, 0, 16, 32\},$$

$$b_j = \{-32, -32, -32, -32, -32, -16, -16, -16, -16, -16, \ldots, 32, 32, 32, 32, 32\}.$$

This function has 25 local minima and one global minimum ($f = 0.998$) if y_1 and y_2 are -32.

Figure 2 shows G_{ij} probabilities of HSS and MCS for Example I with $ND = 10$. Because there are 10 elements in **y**, the ideal probability for selecting any element is 0.1, and the value of a selected element can be randomly bumped up or down with a probability of 0.5. Thus the dotted lines in this figure show ideal two-dimensional G_{ij} probabilities, while circle and cross symbols are for the actual generation probabilities from HSS and MCS, respectively. Since HSS can generate more uniform samples in the multivariate space, G_{ij} probability, generated using HSS, is closer to the ideal probabilities than the results from MCS. In contrast, G_{ij} generated from MCS has a high deviation from the ideal probabilities, and hence, MCS requires a large number of moves to approximate the ideal probabilities.

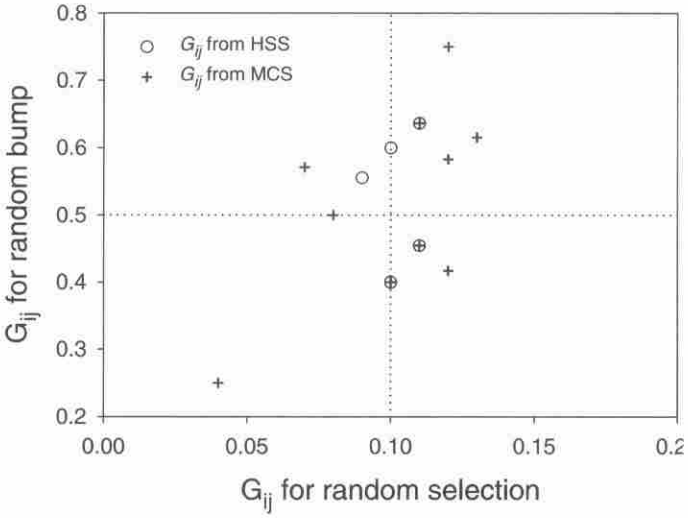

Fig. 2. Generation probabilities G_{ij} for HSS and MCS, Example I with $ND = 10$ (note that several points for HSS overlap).

Figure 3 shows trajectories of the objective value for Example I with different Markov chain lengths. ESA found the global solution with a Markov chain length of 45 at each temperature while the traditional SA exploited a Markov chain length of 75 to reach the same solution. As can be seen, ESA provides a significant reduction in moves at each temperature. Table 2 presents the efficiency improvements of the ESA algorithm in terms of the total

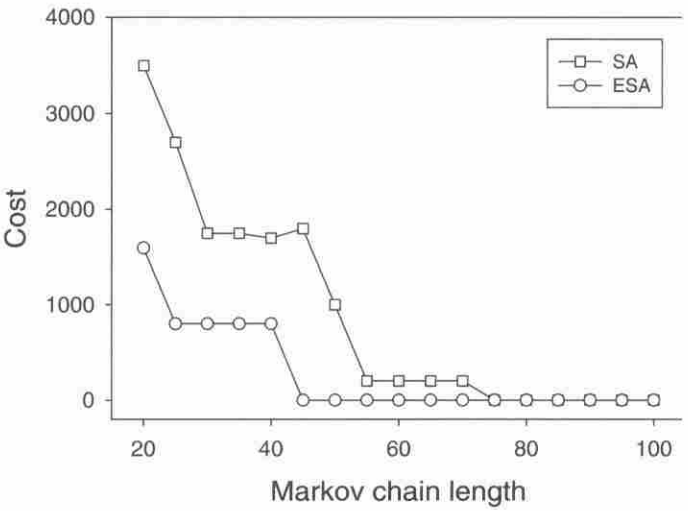

Fig. 3. Cost trajectories of ESA and SA as a function of Markov chain length ($ND = 10$).

Table 2. A comparison of SA and ESA in terms of the total number of moves

Example	*ND*	*Total move (SA)*	*Total move (ESA)*	*Percentage saving (%)*
I	10	3109	1536	50.6
	50	18 598	12 995	30.1
	100	39 790	23 000	42.0
II		1285	592	54.0
III	10	2742	1700	38.0
	40	13 238	6985	47.2
IV		11 625	8075	30.5

number of configurational moves. The results obtained here are the average values for 10 different initial conditions used. From this table, it can be said that ESA is approximately 30–54% more efficient than SA.

3.2. *Extended STochastic Annealing (ESTA)*

Stochastic annealing, a variant of simulated annealing, is designed to solve discrete optimization problems under uncertainty recently proposed by Painton and Diwekar (1995), and Chaudhuri and Diwekar (1996, 1999). This algorithm is designed to efficiently optimize a probabilistic objective function by balancing the solution accuracy and computational efficiency. In the stochastic annealing algorithm, the optimizer obtains not only the decision variables, but also the number of samples required for the stochastic model. Stochastic annealing uses a substitute objective function, which involves the true value of the probabilistic objective function augmented by a penalty term involving an error in sampling related to the number of samples. Thus the stochastic programming problem (ℙ3) can be modified to a new one (ℙ4) that consists of the expected cost function $E_\xi f(\boldsymbol{x}, \boldsymbol{\xi})$ and the penalty term $b(t)\epsilon$, and is given by:

$$(\mathbb{P}4) \quad \min z = E_\xi f(\mathbf{x}, \boldsymbol{\xi}) + b(t)\epsilon, \\ = \frac{1}{N_{\text{samp}}} \sum_{j=1}^{N_{\text{samp}}} f(\boldsymbol{x}, \boldsymbol{\xi}^j) + b(t)\epsilon. \tag{12}$$

The penalty function is composed of the weighting function $b(t)$ and the error bandwidth (confidence interval) ϵ of the sampling method. The weighting function $b(t)$, governed by the cooling schedule, can be expressed in terms of the annealing temperature level (t). At high temperatures, the sample size (N_{samp}) can be small since the algorithm is exploring the functional topology in the configuration space. Thus at this stage computational efficiency is more important than solution accuracy. 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. Based on these properties, an exponential function for $b(t)$ can be devised as:

$$b(t) = \frac{b_0}{k^t}, \tag{13}$$

where b_0 is a small constant (e.g., 0.001) and k is a constant (e.g., 0.92) which governs the rate of increase. These two empirical parameters depend on the cooling schedule and must be predetermined through experimentation such that the penalty term is less than 5% of the real objective function.

The error bandwidth of *random* samples can be estimated from the following equation based on classic statistical methods. No matter what the distribution of X is, the central limit theorem allows one to calculate the probabilistic error bands on the expected value of the random samples generated by a simple MCS. For a 95% confidence interval, the standard normal variate is approximately equal to a value of constant two resulting in a following equation.

$$\epsilon_{\text{MCS}} = \frac{2\sigma}{\sqrt{N_{\text{samp}}}}, \tag{14}$$

where σ is a standard deviation.

In our first approach for the development of stochastic annealing algorithms, we have used this error bandwidth in the penalty function, allowing stochastic annealing to control the number of samples.

Main steps of the STA algorithm are summarized in Table 3. A new configuration S' is generated from the current configuration S based on the given generation probability G_{ij} (Step 2.1.1. and 2.1.2.). N_{samp} in Step 2.2. can be randomly increased or decreased but eventually is governed by the weighting function used in the penalty term. After determining and generating N_{samp} uncertain samples, the model runs N_{samp} times with different uncertain parameters to find $E_\xi f(\mathbf{x}, \boldsymbol{\xi})$. The new stochastic objective function shown in Step 2.7. is then used to evaluate the effect of uncertainties on the optimization problems. The remaining steps are the same as the simulated annealing algorithm steps.

The idea for reducing Markov chain length by using HSS, as done in ESA, is exploited here for the new stochastic annealing, algorithm. Additionally, HSS is used in the inner sampling loop for uncertainty analysis. This is likely to reduce the number of samples needed to calculate the objective function accurately. We call this two-level improved stochastic annealing algorithm the Extended STochastic Annealing (ESTA) algorithm.

Consider the difficult deterministic MINLP problem given in Equation (10) that incorporates uncertainties $(\boldsymbol{\xi})$ in this problem results in the following objective function, a stochastic MINLP to be minimized.

$$f(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) = \sum_{i=1}^{ND} \left(\xi_i x_i - \frac{i}{ND} \right)^2 + \sum_{i=1}^{ND} (\xi_i y_i^2) - \prod_{i=1}^{ND} \cos(4\pi \xi_i y_i). \tag{15}$$

Table 3. Main steps in the stochastic annealing algorithm

Step 1. Initialize variables: T_{initial}, T_{freeze}, α, accept and reject limits, and initial configuration (S)
Step 2. If ($T > T_{\text{freeze}}$), then perform the following loop N (number of moves at a given temperature) times.
2.1. Generate a move S' from the current configuration S as follows:
2.1.1. Select the decision variables for new configuration S' (zero-one, integer, discrete, and continuous variables).
2.1.2. Select the parameter level *randomly* within the neighborhood.
2.2. Select the number of samples N_{samp} by a random move. If $rand(0, 1) \leq 0.5$, then
$N_{\text{samp}} = N_{\text{samp}} + 5 \times rand(0, 1)$
else
$N_{\text{samp}} = N_{\text{samp}} - 5 \times rand(0, 1)$
2.3. Generate N_{samp} samples of the uncertain parameters.
2.4. Perform the following loop N_{samp} times.
2.4.1. Run the model.
2.4.2. Calculate the objective function cost, $f(\mathbf{x}; \boldsymbol{\xi})$.
2.5. Evaluate the expected value $E_{\xi} f(\mathbf{x}, \boldsymbol{\xi})$ and the variance of the cost function.
2.6. Generate the weighting function $b(t) = b_o/k^t$
2.7. Calculate the modified objective function:
$z(S') = \frac{1}{N_{\text{samp}}} \sum_{j=1}^{N_{\text{amp}}} f(\mathbf{x}, \boldsymbol{\xi}^j) + b(t) \frac{2\sigma}{N_{\text{amp}}^{0.5}}$
2.8. Let $\Delta H = z(S') - z(S)$.
2.9. If $\Delta H \leq 0$, then accept the move and set $S = S'$. Else if ($\Delta H > 0$), accept with a probability, exp $(-\Delta H/T)$.
Step 3. Set $T = \alpha T$. If ($T > T_{\text{freeze}}$), return to Step 2.
Step 4. Stop.

Since this example problem has a continuous decision vector ($\mathbf{x}$) and a discrete decision vector ($\mathbf{y}$), a coupling of STA and NLP (STA-NLP), similar to SA-NLP, is used.

Table 4 shows the total NLP subprogram calls using the STA and the new ESTA algorithms. Even for the same Markov chain length (the first two rows), ESTA is approximately 45% more efficient than the STA approach. This reduction is mainly attributed to the faster convergence property of the HSS technique. Further reduction can be achieved when a minimum Markov chain length for each STA is determined (the last row). To find a minimum Markov chain length, the concept of cost of epoch proposed by (Skiscim and Golden (1983) is applied to determine a *pseudo*-thermal-equilibrium at each temperature level. A current epoch (summation of a certain consecutive cost values) is compared with the previous epoch, and if the difference between epoches is within tolerance, then we assume *pseudo*-thermal-equilibrium. The minimum Markov chain lengths at each temperature level were 60 for STA and 40 for ESTA, and hence the efficiency improvement reaches up to 65%. This reduction is a combined effect of the uniformity property of the HSS-based random number generator and the faster convergence property of HSS.

Table 4. A comparison of STA and ESTA in terms of the average number of NLP subprogram calls

ND	*STA*	*ESTA*	*Percentage savings (%)*	*Markov chain length*
2	6218	3298	47.1	100
10	5670	3265	42.4	100
10	4015	1439	64.2	Minimum

3.3. *Hammersley STochastic Annealing (HSTA)*

The error bandwidth used in the STA and ESTA algorithms presented in the previous section are based on classic statistical methods (Equation (14)). Classical statistical methods provide good estimates for the bounds (confidence intervals) of the Monte Carlo sampling but may not be applicable to other less random, yet uniform, sampling techniques. It has been shown that classic statistical methods used to characterize the error bandwidth for any confidence level of HSS overestimate either the confidence intervals or bounds (Chaudhuri and Diwekar, 1999; Diwekar, 2000). Hence, the combination of the HSS technique and the classic error bandwidth (ϵ_{MCS}) in the ESTA algorithm is not the most efficient approach, and further improvement is made possible by using a modified error bandwidth that is specific to the HSS technique.

An approach to quantify the error bandwidth for any probabilistic function is clearly outlined in Chaudhuri and Diwekar (1999) and Diwekar (2000) and is based on concepts from fractal geometry. In this earlier work, it was established that the relative error bandwidth of the Monte Carlo and HSS techniques shows a scaling relationship (N_{samp}^d) with respect to the number of samples. For the HSS technique, the exponent for the error bandwidth for the mean is found to be –1.8 from com-

prehensive simulations, and thus the HSS-specific error bandwidth is given by:

$$\epsilon_{\text{HSS}} \propto \frac{1}{N_{\text{samp}}^{1.8}}. \tag{16}$$

Note that the d of the MCS technique is estimated as -0.5. That is the exactly same value obtained from the classic statistical methods as shown in Equation (14). Finally, the stochastic programming problem (ℙ4) becomes:

$$(\mathbb{P}5) \quad \min\ z = \frac{1}{N_{\text{samp}}} \sum_{j=1}^{N_{\text{samp}}} f(\boldsymbol{x}, \boldsymbol{\xi}^j) + b(t) \times \frac{1}{N_{\text{samp}}^{1.8}}. \tag{17}$$

A new variant of stochastic annealing, HSTA (Hammersley STochastic Annealing), therefore, incorporates: (i) HSS for the generation probability G_{ij}; (ii) HSS in the inner sampling loop for N_{samp} determination; and (iii) the HSS-specific error bandwidth in the penalty term to solve the stochastic programming problem (ℙ5). To evaluate the efficiency improvement by this new HSTA algorithm, the same probabilistic objective equation (Equation (15)) is used.

Figure 4(a and b) shows trajectories of N_{samp} and the penalty term in a percentage of the objective function for the ESTA and HSTA algorithms. From this figure we can see that N_{samp} is significantly reduced when stochastic annealing-based algorithms are used. Further decrease in N_{samp} is observed for the HSTA algorithm due to reduced error bandwidth. Note that there is a large difference in the penalty percentage values between the ESTA and the HSTA. Table 5 shows results comparing the hierarchical improvements from stochastic optimization with fixed N_{samp} to the newest HSTA algorithm. In the table, problem (ℙ3) is the stochastic optimization with fixed N_{samp} while the problems (ℙ4)/(ℙ5) are the stochastic optimization with automatically varying N_{samp}. The fixed N_{samp} is 100 in this comparison while the N_{samp} at (ℙ4)/(ℙ5) algorithms spans from 15 to 90. The base case for comparison is SA with a fixed N_{samp}, in which the objective function in SA is modified to (ℙ3). The base case requires

Table 5. A comparison of the levels of algorithm improvements (ND = 10)

Algorithm	*Problem*	*Total moves*	*Percentage savings (%)*
SA + fixed N_{samp}	(ℙ3)	274 200	–
ESA + fixed N_{samp}	(ℙ3)	170 000	38.0
STA	(ℙ4)	5670	97.9
ESTA	(ℙ4)	3265	98.8
HSTA	(ℙ5)	1793	99.3

Fixed N_{samp} = 100.

274 000 moves to find the optimal solution $\mathbf{x}^*$. This stochastic programming can be improved up to four levels: ESA with fixed N_{samp}, STA, ESTA, and HSTA. A large improvement in computational efficiency can be achieved if ESA is used instead of SA. We can further see that there is significant improvement when we change the problem type from (ℙ3) to (ℙ4)/(ℙ5). The improvement is over 97%, and is mainly attributed to the use of the penalty term and the properties of the HSS sampling technique. Among the stochastic annealing algorithms, HSTA is 68% faster than the basic STA algorithm.

The following real world case study illustrates the usefulness of this approach for large-scale combinatorial stochastic programming problems.

4. Case study: solvent selection

Solvents are extensively used as process materials (e.g., extracting agent) or process fluids (e.g., CFC) in chemical process industries, pharmaceutical industries, and solvent-based industries (such as coating and painting). Since waste solvents are a main source of pollution to air, water, and soil, it is desirable to use reduced amounts of solvents and/or environmentally friendly solvents without sacrificing process performance. There are some solvents that must be eliminated because of environmental and health effects and regulatory requirements. For example, the Montreal Protocol bans many chlorinated solvents.

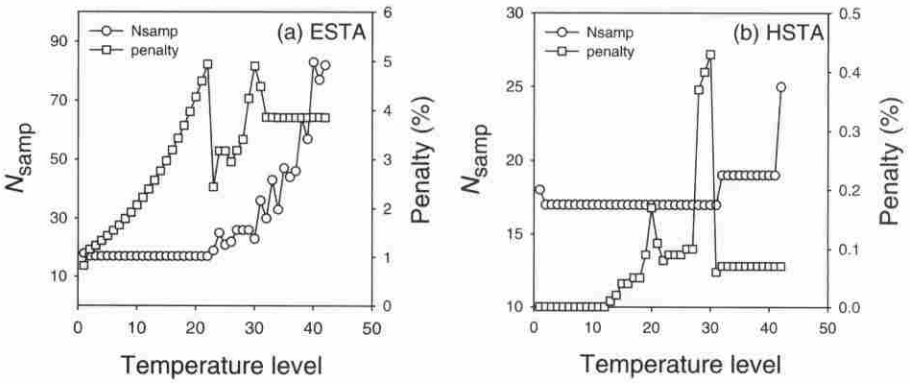

Fig. 4. The trajectories of N_{samp} and penalty percentage of the: (a) ESTA algorithm; and (b) the HSTA algorithm.

Solvent selection, an approach used to generate candidate solvents with desirable properties, can help to handle these problems. Several methodologies have been developed for solvent selection over the years. The first approach uses traditional laboratory synthesis and test methodology to find promising solvents. This method can provide reliable and accurate results, but in many cases is limited by cost, safety, and time constraints. The second approach is to screen the property database. Although it is the most common and simple method, it is limited by the size and accuracy of the database. However, these two methods may not provide the best solvent because of the sheer number of solvent molecules to be tested or searched. Finally, Computer-Aided Molecular Design (CAMD) can automatically generate promising solvents from their fundamental building blocks. CAMD is generally the *reverse* use of the group contribution method that is used to generate molecules having desirable properties. A basic diagram of CAMD is shown in Fig. 5, in which there is a set of groups as a starting point. These groups are uniquely designed to generate all possible molecules by exploring all possible combinations. The properties of each group and/or the interaction parameters between groups can be theoretically calculated, experimentally obtained, or statistically regressed. From this set of groups, solvent molecules can be generated by group combinations. For example, ethanol (CH_3-CH_2OH) is generated from the CH_3, CH_2, and OH groups. Constraints from physical and chemical properties, as well as those from regulatory restrictions, may be imposed, and hence the number of combinations can be reduced. Once molecules are generated, the properties of the molecules are predicted based on the properties of their groups in order to determine if they satisfy the specified criteria. This method can generate lists of candidate solvents with reasonable accuracy within moderate time scale. However, CAMD is limited by the availability and reliability of property estimation methods. All methodologies for solvent selection are exposed to uncertainties that arise from experimental errors, imperfect theories or models and their parameters, or the improper knowledge or ignorance of systems. Although uncertainties can affect the real implementation of selected solvents, few papers in the literature have focused on uncertainties. In this study the new framework is applied to generate greener solvents for acetic acid extraction from water as a case study. The problem is to find the best solvents for acetic acid extraction from water.

Solvent selection is an automatic generation of solvent molecules with desired properties. Computer-Aided Molecular Design (CAMD) constructs candidate solvent molecules from their building blocks or groups and estimates properties of the generated solvents. Then the stochastic programming algorithm based on ($\mathbb{P}5$) is used to select groups to obtain solvent molecules, propagate N_{samp} samples, and optimize the problem to find the best solvent. Since the properties of the building blocks are inherently uncertain and there are large number of molecular combinations of groups, this case study provides a good test for the newly developed combinatorial stochastic optimization algorithm, the HSTA.

4.1. *Solvent selection model*

To replace the current solvent or design a new one, there are several criteria such as distribution coefficient (m), solvent selectivity (β), solvent loss (S_L), and physical properties like boiling point, density, viscosity, and so on that must be satisfied. Of these, the distribution coefficient (m), a measure of solvent capacity, is the most important factor and is defined as:

$$m = \frac{\gamma^{\infty}_{B,A}}{\gamma^{\infty}_{B,S}}, \tag{18}$$

where the symbols, A, B, and S, represent a nonpolluting molecule (e.g., water), polluting molecule (e.g., acetic acid), and solvent molecule (e.g., ethanol), respectively.

In the above equation, the distribution coefficient is a function of the infinite dilution activity coefficients (γ^{∞}), which shows the non-ideality of the mixtures (A–B, A–S, and B–S). If the mixture is ideal, γ^{∞} is close to one. Otherwise, γ^{∞} tends to be greater than one or close to zero. γ^{∞} itself is a function of *groups*, temperature (T), pressure (p), and concentration (C_i). There are several group contribution methods for the prediction of γ^{∞}. The most popular method is the UNIFAC group contribution

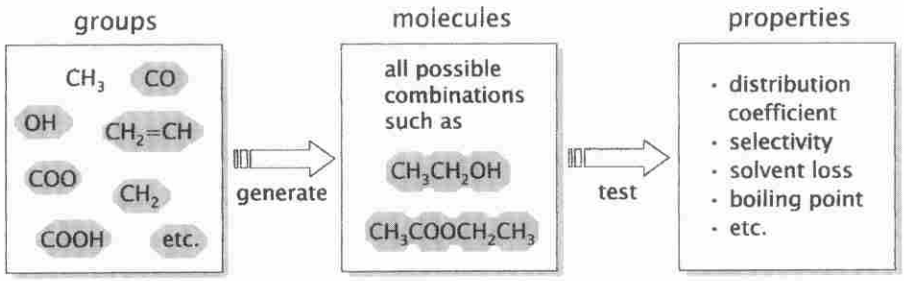

Fig. 5. A basic diagram of CAMD based on group contribution methods.

method (Hansen *et al.*, 1991). In the UNIFAC method, the activity coefficient (γ_i) of a molecule i has two parts: the combinatorial and residual part.

$$\ln \gamma i = \ln \gamma_i^{\mathrm{C}} + \ln \gamma_i^{\mathrm{R}}. \tag{19}$$

The combinatorial part reflects the volume and surface area of each molecule; hence, the volume (q_k) and surface area (r_k) parameters of each group k in molecule i are involved to estimate the combinatorial part. The residual part represents the interaction energies of the molecules; hence, the volume parameter and interaction parameters (a_{mn}, a_{nm}) between groups m and n in the mixture are required to predict γ_i^{R}. Two interaction parameters can be obtained by the regression of experimental data of the mixture. The infinite dilution activity coefficient of molecule i in a mixture is a limiting activity coefficient when the concentration (C_i) of molecule i in a mixture tends to zero, as shown in Equation (20) below. In short, γ_i^{∞} is a function of volume and surface area parameters, interaction parameters, temperature, pressure, and concentration. A detailed UNIFAC model equation is given in Appendix A.

$$\gamma_i^{\infty} = \lim_{C_i \to 0} \gamma_i(q, r, a, T, p, C_i). \tag{20}$$

Even though the UNIFAC group contribution method has several limitations, it provides a useful model for screening and guiding solvent molecules.

The deterministic solvent selection problem is given below:

$$\min \quad -\frac{\gamma_{\mathrm{B,A}}^{\infty}(N_1, N_2^{(i)})}{\gamma_{\mathrm{B,S}}^{\infty}(N_1, N_2^{(i)})}, \tag{21}$$

subject to

$$\beta(\text{solvent selectivity}) = \frac{\gamma_{\mathrm{A,S}}^{\infty}(N_1, N_2^{(i)})}{\gamma_{\mathrm{B,S}}^{\infty}(N_1, N_2^{(i)})} \geq 7,$$

$$S_{\mathrm{L}}(\text{solvent loss}) = \frac{1}{\gamma_{\mathrm{S,A}}^{\infty}(N_1, N_2^{(i)})} \leq 0.058,$$

$$47\,^{\circ}\mathrm{C} \leq T_{\mathrm{BP}}(\text{boiling point}) = \sum_{i=1}^{N_1} t_a \times N_2^{(i)} + t_b \leq 108\,^{\circ}\mathrm{C},$$

$$2 \leq N_1 \leq 10,$$

$$1 \leq N_2^{(i)} \leq 24, \quad \forall i \in N_1.$$

In this problem, the discrete decision variables are the number of groups (N_1) in a solvent molecule and the group index ($N_2^{(i)}, i = 1, \ldots, N_1$) of that molecule. These groups can then build a unique solvent molecule that represents a configuration in the SA-based algorithms. The UNIFAC model is used to predict the objective function, and other solvent selection criteria such as solvent selectivity (β) and solvent loss (S_{L}). Solvent selectivity, the ability of the solvent to selectively dissolve a polluting molecule, and solvent loss, the measure of the solvent loss tendency, are also functions of γ^{∞}. To estimate boiling points (T_{BP}) as a constraint, the boiling point prediction group contribution equation has t_a and t_b as parameters. The bounds on the constraints are taken from the properties of the current solvent for acetic acid extraction.

The group index used in this case study is summarized in Table 6. Since the total number of groups is 24 and a maximum of 10 groups per molecule is allowed, the total combinatorial space is composed of 24^{10} (6.34×10^{13}) combinations. The three UNIFAC parameters, surface area, volume, and interaction parameters, as well as boiling point parameters are tabulated in Appendix B. This represents the deterministic IP problem which becomes a difficult stochastic programming problem when uncertainties are included in the formulation.

4.2. *Uncertainty quantification*

Uncertainties are ubiquitous and inevitable in any processes or systems. They arise from imperfect theories or models and their parameters, and improper knowledge or ignorance of processes. Because these uncertainties are directly linked to economic loss, we want to understand uncertainties, figure out key decision parameters, and try to minimize the effects of these uncertainties.

The group parameters in Equation (18) that will be used in the UNIFAC equation, have three terms: the surface area (R_k) and volume (Q_k) of each group, and the interaction parameters between groups (a_{mn}). The surface

Table 6. A set of discrete decision variables

i	$N_2^{(i)}$	i	$N_2^{(i)}$	i	$N_2^{(i)}$	i	$N_2^{(i)}$
1	CH_3-	2	$-CH_2-$	3	$-CH<$	4	$>C<$
5	$CH_2=CH-$	6	$-CH=CH-$	7	$CH_2=C<$	8	$-CH=C<$
9	$>C=C<$	10	$-OH$	11	CH_3OH	12	H_2O
13	CH_3CO-	14	$-CH_2CO-$	15	$-CHO$	16	CH_3COO-
17	$-CH_2COO-$	18	$HCOO-$	19	CH_3O-	20	$-CH_2O-$
21	$>CH-O-$	22	$-COOH$	23	$HCOOH$	24	$-COO-$

$-$, $>$, and $<$ represent the number of connecting nodes for molecular combinations.

area and volume of each group are constants because they are calculated from atomic and molecular structure data. However, the interaction parameters are obtained from the regression of huge experimental data and thus are subject to uncertainties due to experimental and regression errors. Further, activity coefficients (γ_i) at a finite condition (i.e., $C_i = 0.000\,01$) are, by definition, extrapolated to infinite dilution activity coefficients ($\gamma_i^\infty = \lim_{C_i \to 0} \gamma_i$) in which large discrepancies between experimental and calculated values can be observed.

Uncertainty quantification involves finding the type of probability distribution, and quantifying the first and second moments (i.e., mean and variance) and higher moments of the distribution. The group contribution UNIFAC equation (Equations (A1)–(A4) in Appendix A) for γ^∞ estimation is a complicated nonlinear logarithmic function. The interaction parameters, which are the main source of uncertainty, are in the deepest part of the UNIFAC equation, and the total number of interaction parameters are close to 2000.

Aside from the mentioned difficulties for uncertainty analysis, the numerical values of the interaction parameters are scattered, and for each binary pair of groups the interaction parameters are *highly correlated* (Xin and Whiting, 2000). Therefore, one cannot elicit all uncertainty information of the interaction parameters nor apply uncertainties on the all interaction parameters due to mathematical and computational complexity.

In this paper, we have used a unique and useful way of representing these complicated uncertainties. Information regarding to the thermodynamic properties is utilized to lump the highly correlated interaction parameters into one parameter (more precisely, one parameter with three categories). We introduce a new uncertainty quantification term called the Uncertainty Factor (UF), which is a ratio of the experimental γ^∞ to the calculated γ^∞. UF is defined in Equation (22) and shows how much the calculated γ^∞ is deviated from the true γ^∞.

$$\mathrm{UF} = \frac{\gamma_{\mathrm{exp}}^\infty}{\gamma_{\mathrm{calc}}^\infty}. \tag{22}$$

Note that a UF of unity means the calculated value is exactly equal to the experimental value.

Further, since the properties of water are quite different from those of organic chemicals, γ^∞ and thus the UF can be divided into three categories based on the mixture systems: γ^∞ in the organic-water, water-organic, and organic-organic systems.

The UFs, the estimated γ^∞s of 227 binary chemical systems are compared with the experimental values obtained from Barton (1983). Figure 6 shows a probability density function (pdf) of the UF of $\gamma_{\mathrm{organic,water}}^\infty$. It can be seen from this figure that the type of distribution on this γ^∞ is a lognormal distribution with the arithmetic mean of 2.92 and the standard deviation of 5.94. From this pdf

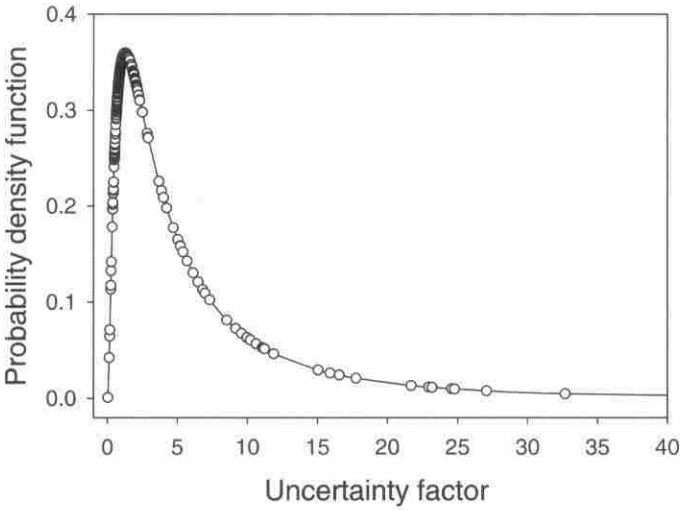

Fig. 6. The probability density function of UF of $\gamma_{\mathrm{organic,water}}^\infty$ (a total of 227 experimental data are compared).

one can expect that uncertainties exert great impact on the γ^∞ due to the large mean and wide standard deviation. It is also found that the probability distribution functions of UF of the other two infinite activity dilution coefficients ($\gamma_{\mathrm{water,organic}}^\infty$ and $\gamma_{\mathrm{organic,organic}}^\infty$) are normally distributed ($N(1.08,\ 0.37)$) and lognormally distributed ($\log N(1.42,\ 1.14)$), respectively. The values of $\gamma_{\mathrm{water,organic}}^\infty$ tend to be less affected by uncertainties because nonideality caused by water is small in these mixtures. It is interesting to see that all the distributions are shifted to the right or are positively skewed. Therefore, uncertainties not only perturb γ^∞ but also increase the output values of γ^∞.

4.3. *The stochastic programming problem and results*

The discrete stochastic programming problem for finding the best candidate solvents for acetic acid extraction from water is then formulated as follows:

$$\min \quad \frac{1}{N_{\mathrm{samp}}} \sum_{j=1}^{N_{\mathrm{samp}}} \left[\frac{\gamma_{\mathrm{B,A}}^\infty(N_1, N_2^{(i)}) \times \xi_1^j}{\gamma_{\mathrm{B,S}}^\infty(N_1, N_2^{(i)}) \times \xi_3^j} \right], \tag{23}$$

subject to

$$\xi_1 \sim \log N(2.92, 5.94),$$
$$\xi_2 \sim N(1.08, 0.37),$$
$$\xi_3 \sim \log N(1.42, 1.14),$$

$$\beta(\text{solvent selectivity}) = \frac{1}{N_{\mathrm{samp}}} \sum_{j=1}^{N_{\mathrm{samp}}} \left[\frac{\gamma_{\mathrm{A,S}}^\infty(N_1, N_2^{(i)}) \times \xi_2^j}{\gamma_{\mathrm{B,S}}^\infty(N_1, N_2^{(i)}) \times \xi_3^j} \right] \geq 7,$$

$$S_{\mathrm{L}}(\text{solvent loss}) = \frac{1}{N_{\mathrm{samp}}} \sum_{j=1}^{N_{\mathrm{samp}}} \left[\frac{1}{\gamma_{\mathrm{S,A}}^\infty(N_1, N_2^{(i)}) \times \xi_1^j} \right] \leq 0.058,$$

$$47^{\circ}\mathrm{C} \le T_{\mathrm{BP}}(\text{boiling point}) = \sum_{i=1}^{N_1} t_a \times N_2^{(i)} + t_b \le 108^{\circ}\mathrm{C},$$

$$2 \le N_1 \le 10,$$

$$1 \le N_2^{(i)} \le 24, \quad \forall i \in N_1,$$

where ξ_i are uncertain parameters of UF_i and imposed on the output γ^{∞}. Discrete decision variables are the number of groups (N_1) in a solvent molecule and the group index $(N_2^{(i)}, i = 1, \ldots, N_1)$ of that molecule that can build a unique solvent molecule that is a configuration in the SA-based algorithms.

To move to a new configuration (i.e., group combination) from the current configuration in the HSTA algorithm (see Step 2.1. in Table 3), there are three processes used: addition, contraction, and random bump. In addition process $(N_1 = N_1 + 1)$ the number of groups (N_1) in a solvent molecule is increased, and a random group index is assigned to that increased group. In contraction process $(N_1 = N_1 - 1)$, one group is randomly deleted. In random bump $(N_1 = N_1)$, the number of groups in a molecule is unchanged. Instead, an arbitrarily selected group index $(N_2^{(i)})$ is randomly bumped up or down. The magnitude of these bumps are also random. The probabilities for these three processes are specified at 30, 30, and 40%, respectively. A large random bump probability is guaranteed to span all the group indexes. Besides these basic probabilities, there are also several probabilities for configurational moves.

Figure 7(a) shows the progress of the HSTA (Hammersley STochastic Annealing). Two objective values are plotted with respect to the annealing temperature level: the true (expected) objective and the objective with penalty term. At the beginning of annealing, exploring the configuration surface to locate local optima is more important than the solution accuracy of these local optima. Therefore, the two objectives are very close because of a small weighing function $b(t)$ that results in a small penalty term. As annealing proceeds, solution accuracy becomes more important than solution efficiency, and $b(t)$ increases very rapidly for not being precise. To maintain or enhance the solution accuracy by reducing the penalty, the trend is to increase number of samples N_{samp}, as shown in Fig. 7(b).

Table 7. Top 15 candidate solvents for both cases

Rank	*Solvents*	*m (distribution coeffcient)*	
		Stochastic	*Deterministic*
1	$2CH_3$, CH_2, CH, HCOO	2.95	0.87
2	CH_3, CH_2, CH=CH, HCOO	2.60	0.75
3	CH_3, CH_2, CH_2=C, HCOO	2.55	0.75
4	CH_3, CH_2=CH, $2CH_2O$	2.27	
5	CH_3, CH_2=CH, CH_3O CH—O	2.27	
6	CH_3, CH, CH_2=CH, CH_3O, CH_2O	2.15	0.63
7	CH_3, CH_2=C, CH_3CO	2.09	0.61
8	CH_3, CH=CH, CH_3CO	2.08	0.61
9	CH_3, CH_2, CH_2=C, CH_3O, CH_2O	2.04	
10	CH_3, $2CH_2$, CH_3CO	1.96	0.60
11	CH_3, CH_2, CH_2=C, CH_3O	1.84	
12	$2CH_2$, CH_2=CH, CH_2O, CH_3O	1.55	
13	CH_3, CH=CH, CHO	1.51	
14	CH_3, CH_2, CH, CH_2=CH, CH_3O	1.49	
15	$2CH_3$, CH, CH_2=C, CH_3O	1.41	

Table 7 shows the optimal solvent candidates for the stochastic (HSTA) and deterministic (STA) cases. Both cases generated 40 candidate solvents, and only the top 15 are ranked with respect to the order in the stochastic case, and thus only seven solvents from the deterministic case

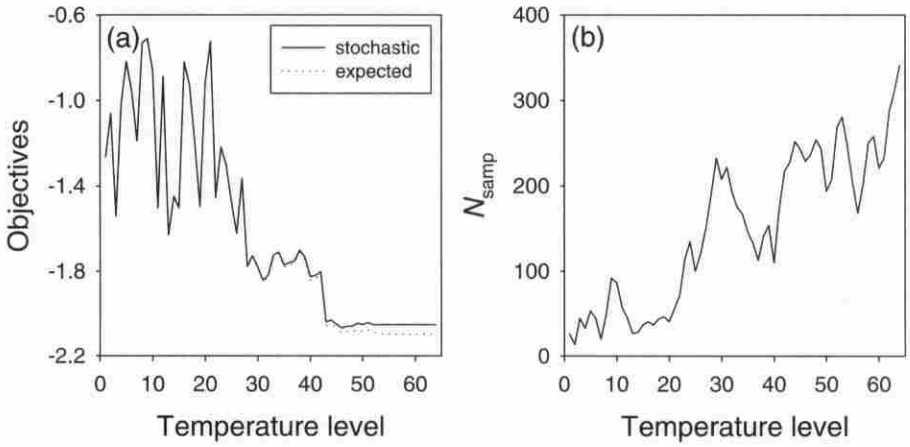

Fig. 7. (a) Changes of the objective values; and (b) changes in the number of samples.

are listed in this table (see the blanks at the deterministic case). This implies that the deterministic case does not generate several or many promising solvents, which appeared in the stochastic case. As expected, distribution coefficients for the stochastic case are greater than those of the deterministic case due to the positively skewed uncertainty factor, mainly $\gamma^{\infty}_{\text{organic,water}}$, and because increased m values become closer to the experimental values.

Table 7 presents these candidate solvents for further screening in terms of design, safety, health, and environmental constraints. Solvent availability, toxicity evaluation, and safety consideration are some valuable criteria that may be used. Finally, experimental verifications should be followed. After screening, if the first top three solvents in Table 7 were not found to be *practically* useful[1], then different candidate solvents between the stochastic and deterministic cases could lead to big differences in the whole solvent selection process. Thus it is clear from this result that real implementation of CAMD without considering uncertainties may fail to find the best solution.

The *Value of Stochastic Solution* (VSS) (Birge and Louveaux, 1997) can be used to quantify the effects of uncertainty. The difference between taking the average value of the uncertain variable as the solution as compared to using the stochastic analysis is defined as VSS. Thus VSS represents the loss by not considering the uncertainties. Because the uncertainty factors represented as ξ over γ^{∞} are implemented in the objective function, we can assume that the expected value of the stochastic problem with the average $\boldsymbol{\xi}$ be 2.06 ($=2.92/1.42$) times the deterministic m shown in Table 7. For the first set of solvent molecules in this table, the VSS of this case study is estimated as $1.16(= 2.95 - 2.06 \times 0.87)$, and the stochastic optimization, therefore, increases the performance (distribution coefficient in this study) by 65%. Other sets of solvent molecules have a similar VSS.

Further, we can also observe differences in the probability density function of the solvents. Figure 8 shows the probability density functions (pdf) of distribution coefficients of the top 40 solvents for each case. The pdf of the deterministic case looks like a narrow lognormal distribution with a small standard deviation and has a strong peak at the distribution coefficient of 0.53. On the contrary, the pdf of the stochastic case is a wide normal distribution with a mean of 1.34 due to positive skewness of the uncertainty factors. The narrow deterministic lognormal pdf is changed to a wide normal pdf and shifted positively under the stochastic case, and the stochastic case can thus cover wider range of the configuration space. In addition, the types of proposed solvents are different from each other. Although most of the solvents are in one of the following types – formates, alcohols, esters, or ethers – the stochastic case can provide additional types of solvents that are alkanes and alkenes. From this case study we can see the importance of uncertainties and the usefulness of the HSTA algorithm for large-scale combinatorial stochastic programming problems.

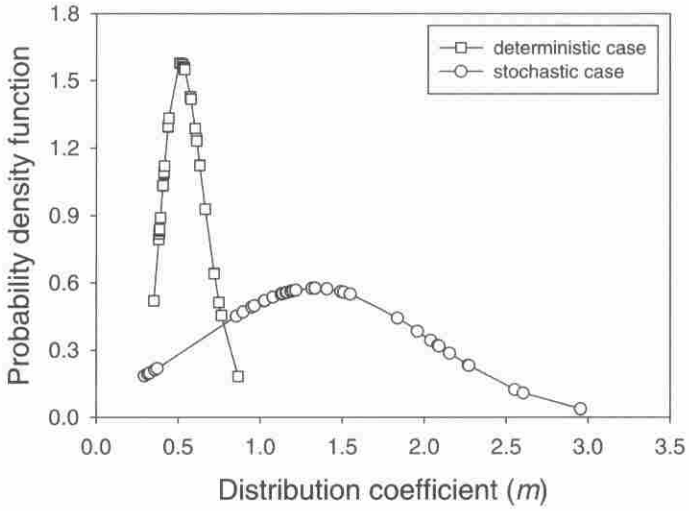

Fig. 8. Probability density functions of distributed coefficients for the deterministic and stochastic cases.

5. Conclusion

This paper presented hierarchical improvements in the SA-based algorithm for solving large-scale discrete optimization problem under uncertainty. At first, the deterministic SA was modified to ESA, which exploited the uniformity property of the HSS technique, and which resulted in a shorter Markov chain length at each temperature level. The same concept was then applied to the stochastic annealing algorithm, resulting in the new ESTA algorithm that provided the trade-off between computational efficiency and solution accuracy by automatically determining N_{samp}. In addition, the faster convergence property of HSS is also incorporated in this algorithm. The efficiency is improved further in the HSTA algorithm by using the HSS-specific error bandwidth in the penalty term of the probabilistic objective functional. For stochastic MINLP problems, HSTA was coupled with the NLP algorithm. For the test problems considered in this study, the combined improvement of all the steps in the hierarchy was shown to result in 99.3% savings in computational time. A real-world look at solvent selection under uncertainty for acetic acid extraction was presented as a case study. It is concluded that the stochastic programming solution provides different sets of candidate solvents with more promising properties as

[1] The generated solvents are chemically possible, but they may be economically infeasible. We cannot estimate the cost of generated solvents using any group contribution methods.

compared to the deterministic solution. Further, the stochastic solution covered a wider range of the configurational surface. Therefore, the HSTA algorithm can be a useful tool for large-scale combinatorial stochastic programming problems.

Acknowledgements

We thank the National Science Foundation for funding this research (Goali Project: CTS-9729074).

References

Barton, A. (1983) *CRC Handbook of Solubility Parameters and Other Cohesion Parameters*, CRC Press, Boca Raton, FL.

Birge, J.R. and Louveaux, F. (1997) *Introduction to Stochastic Programming*, Springer, New York, NY.

Chaudhuri, P. and Diwekar, U.M. (1996) Process synthesis under uncertainty: a penalty function approach. *AIChE Journal*, **42**(3), 742–752.

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

Dantzig, G. and Glynn, P. (1990) Parallel processors for planning under uncertainty. *Annals of Operations Research*, **22**, 1–21.

De Jong, K.A. (1981) An analysis of the behavior of a class of genetic adaptive system. Ph.D. thesis, University of Michigan, Ann Arbor, MI.

Diwekar, U.M. (2000) An efficient approach to optimization under uncertainty, *Computational Optimization and Applications* (accepted).

Hammersley, J.M. (1960) Monte Carlo methods for solving multivariate problems. *Annals of the New York Academy of Science*, **86**, 844–874.

Hansen, H., Rasmussen, P., Fredenslund, A., Schiller, M. and Gmehling, J. (1991) Vapor-liquid equilibria by UNIFAC group contribution: 5. Revision and extension. *Industrial and Engineering Chemistry Research*, **30**, 2352–2355.

Hansen, H.K. (1991) UNIFAC manual 8101. Technical report, The Technical University of Denmark, Lyngby, Denmark.

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

Iman, R.L. and Conover, W.J. (1982) Small-sample sensitivity analysis techniques for computer models, with an application to risk assessment, *Communications in Statistics – Part A, Theory and Methods*, **17**, 1749–1842.

Joback, K.G. (1984) Master's thesis, Department of Chemical Engineering, MIT, Cambridge, MA.

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

Kirkpatrick, S., Gelatt, Jr, C.D. and Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

Knuth, D.E. (1973) *The Art of Computer Programming: Fundamental Algorithms*, Vol. 1, Addison-Wesley, Reading, MA.

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

Narayan, V., Diwekar, U. and Hoza, M. (1996) Synthesizing optimal waste blends. *Industrial and Engineering Chemistry Research*, **35**, 3519–3527.

Painton, L.A. and Diwekar, U.M. (1994) Synthesizing optimal design configurations for a Brayton cycle power plant. *Computers and Chemical Engineering*, **18**, 369–381.

Painton, L.A. and Diwekar, U.M. (1995) Stochastic annealing for synthesis under uncertainty. *European Journal of Operational Research*, **83**, 489–502.

Salazar, R. and Toral, R. (1997) Simulated annealing using hybrid Monte Carlo. *Journal of Statistical Physics*, **89**(5/6), 1047–1060.

Shapiro, A. and Homem-De-Mello, T. (2000) On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *INFORMS Journal on Optimization*, **11**(1), 70–86.

Skiscim, C.C. and Golden, B.L. (1983) Optimization by simulated annealing: a preliminary computational study for the TSP, in *Winter Simulation Conference Proceedings* (Arlington, VA), IEEE, New York, pp. 523–535.

Szu, H. and Hartley, R. (1987) Fast simulated annealing. *Physics Letters A*, **3**(14), 157–162.

Van Laarhoven, P. and Aarts, E. (1987) *Simulated Annealing: Theory and Applications*, Reidel Publishing Company, Dordrecht, The Netherlands.

Xin, Y. and Whiting, W.B. (2000) Case studies of computer-aided design sensitity to thermodynamics data and models. *Industrial and Engineering Chemistry Research*, **39**, 2998–3006.

Appendices

Appendix A: The UNIFAC model and example

Activity coefficient (γ_i) of a molecule i has two parts: the combinatorial and residual part (Hansen *et al.*, 1991).

$$\ln \gamma_i = \ln \gamma_i^{\mathrm{C}} + \ln \gamma_i^{\mathrm{R}}. \tag{A1}$$

The combinatorial part reflects the volume and surface area of each molecule; hence, the volume (Q_k) and surface area (R_k) parameters of each group k in the mixture are used to estimate the combinatorial part as given by:

$$\ln \gamma_i^{\mathrm{C}} = \ln \frac{\phi_i}{x_i} + 5q_i \ln \frac{\theta_i}{\phi_i} + l_i - \frac{\phi_i}{x_i} \sum_j x_j l_j, \tag{A2}$$

where

$$\begin{aligned}
l_i &= 5(r_i - q_i) - (r_i - 1);\\
\theta_i &= \frac{q_i x_i}{\sum_j q_j x_j}, \quad \phi_i = \frac{r_i x_i}{\sum_j r_j x_j};\\
q_i &- \textstyle\sum_k v_k^{(i)} Q_k, \quad r_i - \sum_k v_k^{(i)} R_k;\\
x_i &= \text{mole fraction of molecule } i \\
&\quad (\text{for pure component}, x_i = 1);\\
v_k &= \text{number of repetitions of molecule } k;\\
i &= \text{molecule } i;\\
j &= \text{number of molecules in a mixture;}\\
k &= \text{number of groups in molecule } i.
\end{aligned}$$

R_k and Q_k, which are tabulated, are the group size and surface volume parameters that can be obtained from atomic and molecular structure data, the Van der Waals group volumes and surfaces. θ_i and ϕ_i are the surface volume fraction and surface area fraction, respectively of molecule i in the mixture.

The residual part is induced by the difference of the residual activity coefficient (Γ) of group k in a mixture and the residual activity coefficient of group k in molecule i, and is defined as:

$$\ln \gamma_i^{\mathrm{R}} = \sum_k v_k^{(i)} [\ln \Gamma_k - \ln \Gamma_k^{(i)}]. \quad \text{(A3)}$$

The residual activity coefficient is defined as follows:

$$\ln \Gamma_k = Q_k \left[1 - \ln \left(\sum_m \theta_m \tau_{mk} \right) - \sum_m \frac{\theta_m \tau_{km}}{\sum_n \theta_n \tau_{nm}} \right], \quad \text{(A4)}$$

where

$$\theta_m = \frac{Q_m X_m}{\sum_n Q_n X_n},$$

$$X_m = \frac{\sum_j v_m^{(i)} x_j}{\sum_j \sum_n v_n^{(j)} x_j},$$

$$\tau_{mn} = \exp\left[-\frac{a_{mn}}{T} \right] \ (T = \text{temperature in K}),$$

k, m, n = all groups in a mixture.

θ_m is the group surface area fraction and X_m is the group fraction.

As an example, activity coefficients of A (H_2O) and B (CH_3COOH: acetic acid) are estimated at $x_A = 0.5$ and 25°C. The R_k and Q_k are shown in the Table A1.

The interaction energy terms are obtained as:

$$a_{mn} = \begin{pmatrix} 0 & 1318 & 663.5 \\ 300 & 0 & -14.09 \\ 315.3 & -66.17 & 0 \end{pmatrix}. \quad \text{(A5)}$$

By using these data into Equations (A1)–(A4), we can find the activity coefficients listed in Table A2.

Table A1. The R_k and Q_k values

k	R_k	Q_k
H_2O	0.9200	1.4000
CH_3	0.9011	0.8480
COOH	1.3013	1.2240

Table A2. The activity coefficients

i	$ln\ \gamma_i^C$	$ln\ \gamma_i^R$	γ_i
H_2O	0.2305	0.0405	1.3112
CH_3COOH	0.1290	−0.0210	1.1141

Table A3. The surface area R_k and volume Q_k values for the UNIFAC equation and boiling point parameters t_a

Main group No.	*Sub-group*	*Sub-group No. (k)*	R_k	Q_k	t_a
1	CH_3—	1	0.9011	0.8480	23.58
	—CH_2—	2	0.6744	0.5400	22.88
	—CH<	3	0.4469	0.2280	21.74
	>C<	4	0.2195	0.0000	18.25
2	CH_2=CH—	5	1.3454	1.1760	43.14
	—CH=CH—	6	1.1167	0.8670	49.92
	CH_2=C<	7	1.1173	0.9880	42.32
	—CH=C<	8	0.8886	0.6760	49.10
	>C=C<	9	0.6605	0.4850	48.28
3	—OH	10	1.0000	1.2000	92.88
4	CH_3OH	11	1.4311	1.4320	116.46
5	H_2O	12	0.9200	1.4000	175.03
6	CH_3CO—	13	1.6724	1.4880	100.33
	—CH_2CO—	14	1.4457	1.1800	99.63
7	—CHO	15	0.9980	0.9480	74.74
8	CH_3COO—	16	1.9031	1.7280	104.68
	—CH_2COO—	17	1.6764	1.4200	103.98
9	HCOO—	18	1.2420	1.1880	84.88
10	CH_3O—	19	1.1450	1.0880	46.00
	—CH_2O—	20	0.9183	0.7800	45.30
	>CH—O—	21	0.6908	0.4680	44.16
11	—COOH	22	1.3013	1.2240	160.8
	HCOOH	23	1.5280	1.5320	175.53
12	—COO—	24	1.3800	1.2000	81.10

Table A4. Interaction parameters (a_{mn}) between main groups m and n

$m\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	86.02	986.50	697.20	1318.00	476.40	677.00	232.10	507.00	251.50	663.50	387.10
2	−35.36	0.00	524.10	787.60	270.60	182.60	448.80	37.85	333.50	214.50	318.90	48.33
3	156.40	457.00	0.00	−137.10	353.50	84.00	−203.60	101.10	267.80	28.06	199.00	190.30
4	16.51	−12.52	249.10	0.00	−181.00	23.39	306.40	−10.72	179.70	−128.60	−202.00	165.70
5	300.00	496.10	−229.10	289.60	0.00	−195.40	−116.00	72.87	0.00	540.50	−14.09	−197.50
6	26.76	42.92	164.50	108.70	472.50	0.00	−37.36	−213.70	−190.40	−103.60	669.40	−18.80
7	505.70	56.30	529.00	−340.20	480.80	128.00	0.00	−110.30	766.00	304.10	497.50	0.00
8	114.80	132.10	245.40	249.60	200.80	372.20	185.10	0.00	−241.80	−235.70	660.20	560.20
9	329.30	110.40	139.40	227.80	0.00	385.40	−236.50	1167.00	0.00	−234.00	−268.10	−122.30
10	83.36	26.51	237.70	238.40	−314.70	191.10	−7.84	461.30	457.30	0.00	664.60	417.00
11	315.30	1264.00	−151.00	339.80	−66.17	−297.80	−165.50	−256.30	193.90	−338.50	0.00	−337.00
12	529.00	1397.00	88.63	171.00	284.40	123.40	577.50	−234.90	145.40	−247.80	1179.00	0.00

Appendix B: Group parameters for the UNIFAC equation

Table A3 shows the data of surface area (R_k) and volume (Q_k) of each group k in the UNIFAC equation. The values are updated by Hansen (1991), and the group indexes are changed for the case study in this paper. The total number of sub-groups (N_2) is 24 while the number of main groups are 12. Each main group consists of similar sub-groups. For example, linear alkyl sub-groups (CH_3, CH_2, CH, and C) are grouped together as "Main group 1".

Table A3 also shows data for boiling point estimation. The boiling point can be estimated by the following equation (Joback, 1984):

$$T_{\mathrm{BP}}(\text{boiling point}, {}^{\circ}\mathrm{C}) = \sum_{i=1}^{N_1} t_a\left(N_2^{(i)}\right) + t_b, \qquad \text{(A6)}$$

$$\text{where } t_b = 198.12$$

Interaction parameters (a_{mn}) between main groups m and n are summarized in Table A4, where the unit is $1/K$. These parameters are subject to uncertainties caused by measurement and regression errors and imperfect models.

Biography

Dr Urmila Diwekar has been a faculty member in Carnegie Mellon University since 1991. In chemical engineering, she has worked extensively in the areas of simulation, design, optimization, control, stochastic modeling, and process synthesis of chemical processes. She has made major contributions to research on batch distillation including authoring the first book on batch distillation and this work is well recognized. Due to her strong research interests in the areas of pollution prevention, sustainability, and designing for environment, she currently has an appointment in the Department of Civil and Environmental Engineering, Carnegie Mellon University, as a full Professer (Research).

Uncertainties are inherent in real world processes. Recognizing this fact, Urmila started working on stochastic modeling, efficient methods for uncertainty analysis, and optimization under uncertainty as early as 1991. The interdisciplinary nature of the field developed into several research collaborations and in 1999 Urmila founded the Center for Uncertain Systems: Tools for Optimization and Uncertainty (CUSTOM). This is a truly multi-disciplinary center that fosters interactions between various industries, national labs, and various academic discipline. Recently she has completed a breadth level book on Applied Optimization that is discipline independent and is going to published by Kluwer as a lead book in their Applied Optimization series.

Urmila has published more than 60 papers in renowned journals, presented 100 or more papers and seminars, and chaired conferences and sessions in national and international meetings. She has been on the executive committee of the Computers and Systems Analysis (CAST) and the environmental divisions of AIChE. Besides academic research, she has also worked with Simulation Sciences Inc. and is consultant to several industries.

Dr Kim is a postdoctoral associate in CUSTOM (Center for Uncertain Systems: Tools for Optimization and Management) and Civil & Environmental Engineering, Carnegie Mellon University. His Ph.D. work was to develop a multiobjective programming framework under uncertainty for integrating simultaneously environmentally benign solvent design and solvent recycling in chemical processes. This new framework applied to real world case studies showed better economic performance and environmental quality. He has published 8 papers and got two AIChE (American Institute of Chemical Engineers) Graduate Research awards. Currently he is working in deriving theoretical sampling error bounds for random samples. This work will significantly affect the efficacy of stochastic optimization and programming. He is a member of AIChE and ACS (American Chemistry Society).