Process Optimization via Simulated Annealing: Application to Network Design

Simulated annealing is a multivariable optimization technique based on the Monte Carlo method used in statistical mechanical studies of condensed systems and follows by drawing an analogy between energy minimization in physical systems and costs minimization in design applications. In this paper, simulated annealing is introduced and reviewed. The utility of the method for optimization of chemical processes is illustrated by applying it to the design of pressure relief header networks and heat exchanger networks.

W. B. Dolan
P. T. Cummings
M. D. LeVan
Department of Chemical Engineering
University of Virginia
Charlottesville, VA 22901

Introduction

Computer-aided chemical process design and optimization increasingly play an important role in reducing the time lag between process innovation and commercial implementation, in ensuring the efficiency and flexibility of new chemical plants, and in improving the operating efficiencies of existing plants.

This paper applies the simulated annealing algorithm to design problems in chemical engineering. Simulated annealing is a general method for treating a broad class of large, multivariable optimization problems. It has found wide application in the physical sciences and engineering, but evidently not previously to problems in chemical engineering.

The simulated annealing algorithm was proposed by Kirkpatrick et al. (1983), who drew an analogy between the cooling of a fluid and the optimization of a complex system. In order for a pure substance to be 'frozen' into a perfect or nearly perfect crystal, it must be annealed by first melting and then cooling very slowly. If the substance is cooled too quickly, a crystal with many defects or a glass will be formed; the resulting structure will be far from the highly ordered, minimum energy, crystalline state. To illustrate the effectiveness of simulated annealing, Kirkpatrick et al. (1983) applied the method to two combinatorially large optimization problems: the traveling salesman problem and the layout of very-large-scale integrated (VLSI) circuit computer chips. For the latter, the placement and wiring of several thousand discrete circuit elements are principal concerns. In general, the shorter the connections are between the elements, the more optimal is the design; however, the circuit elements cannot be placed too close together since they can interfere with each other. Simulated annealing has now become a standard technique for designing VLSI chips. In particular, it is often used to assess whether designs obtained by other, perhaps heuristic, methods can be improved upon by starting the annealing at relatively low temperature using the proposed design as the initial configuration.

Simulated annealing is based on the theory of Markov chains, as described below. It accepts and rejects randomly generated 'moves' on the basis of a probability related to an 'annealing temperature'. It can accept moves which change the value of an objective function in the direction opposite to that of the desired long-term trend. Thus, for a global minimization problem, a move that increases the value of the objective function (an 'uphill' move) may be accepted as part of the full series of moves for which the general trend is to decrease the value of the objective function. In this way, simulated annealing is able to explore the full solution space and solutions are independent of the starting point. This means that one does not become trapped in a far from optimal local minimum as is possible with algorithms based on Newton's method.

Simulated annealing has proven to be a practical method for solving combinatorially large optimization problems. Press et al. (1986) report that simulated annealing can obtain solutions to combinatorial optimization problems within specified 'distances' of the global optimum in computational time that is proportional to N^n, where N is a measure of the size of the problem and n is a small positive number. For example, Aarts and van Laarhoven (1985) report a simulated annealing method for the N-city traveling salesman problem that yields solutions within 2% of the global optimum and grows as N^3. Algorithms which are *guaranteed* to find the global optimum for such problems increase in computational time much more quickly and, in the worst case, are exponential in N.

With simulated annealing, the minimization of a global objective function is approached directly. When applied to a problem of cost minimization, moves are accepted and rejected based on a cost function. No heuristic arguments are needed.

Instead, both capital costs and operating expenses are contained in the cost function and its global minimum is sought. If, for some reason, constraints are to be built into the problem, this can be easily done. Any move which would violate a constraint can be rejected outright or penalized through the cost function.

One of the great strengths of simulated annealing is the ease with which discrete variables and discontinuous cost functions can be handled. Off-the-shelf supplies and equipment are often available in standard sizes and capacities. Construction costs can be reduced significantly if special fabrication is not required. For example, the cost of a standard heat exchanger is roughly 50 to 75% of that for a custom-built exchanger (Rubin et al., 1984). Similarly, pipes are readily available only in standard sizes. Thus, with simulated annealing, in contrast to many popular algorithms, it is not necessary to solve a problem in terms of continuous variables, only later to have to replace them with standard sizes.

As explained below in detail, we have adopted terminology of statistical mechanics to classify problems into two types: 'canonical' problems, for which the number of elements is fixed, and 'grand canonical' problems, for which the number of elements in the problem varies as the solution is obtained. To our knowledge, problems treated previously using simulated annealing have all been of the canonical type. Thus, our consideration of grand canonical problems is a contribution to the development of simulated annealing as a general methodology for multivariable optimization.

Two types of problems are considered in this paper: the design of a pressure relief header network (a canonical problem) and the design of a heat exchanger network (a grand canonical problem).

Theory

Simulated annealing is based on the Monte Carlo simulation technique developed by Metropolis et al. (1953) to study the statistical mechanics of condensed systems (dense gases, liquids, and solids). More precisely, the Metropolis algorithm permits the direct simulation of the so-called canonical ensemble of statistical mechanics, in which the number of molecules N, the volume V, and the temperature T of the system are held fixed. In so doing, it asymptotically generates low-energy configurations of the system at fixed N, V, and T. The variables that can be manipulated to generate these low-energy configurations are the positions and orientations of the molecules.

In this section, the Metropolis algorithm for the simulation of condensed physical systems is reviewed. In addition, the algorithm for simulating the so-called grand canonical ensemble of statistical mechanics (in which V, T and the chemical potential μ_i of each species are held fixed and N is not prescribed but rather is determined in the course of the simulation) is described. Finally, the simulated annealing algorithm is presented.

Monte Carlo simulation of condensed systems

The mathematics underlying the Metropolis algorithm are given elsewhere (Hammersley and Handscomb, 1964; Valleau and Whittington, 1977) and are beyond the scope of this paper. For simplicity, the Metropolis algorithm will be described for a pure fluid composed of spherically symmetric molecules.

In Monte Carlo simulations of condensed systems, the configurational energy E, which the molecules possess by virtue of their positions and which is calculated from the intermolecular potentials, is determined. E is given completely by the set of $\{\vec{r}_i\}$, $i = 1, \ldots, N$ where $\vec{r}_i$, is the position of molecule i in the system. Equivalently, one can define the configuration state of a system of N molecules by the *phase vector* $\vec{\Gamma}^N = (\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$, a vector in $3N$-dimensional *phase space*. The energy E is clearly a function of $\vec{\Gamma}^N$,

$$E = E(\vec{\Gamma}^N) \tag{1}$$

The object of the Metropolis algorithm is to generate a series of points in phase space $\vec{\Gamma}_i^N$ which are distributed according to the canonical ensemble Boltzmann probability density

$$P(\vec{\Gamma}^N) \propto \exp\left(-E(\vec{\Gamma}^N)/k_B T\right) \tag{2}$$

where k_B is Boltzmann's constant.

The Metropolis algorithm is a rigorously correct formalism for generating a Markov chain (Karlin, 1973) of points in phase space that asymptotically are distributed according to Eq. 2 and are independent of the initial point. The algorithm for generating a sequence of points in phase space is given by the following steps:

(i) Initialize the positions of the molecules (i.e., initialize $\vec{\Gamma}^N$).

(ii) Choose a molecule at random and move it randomly to a new position (randomly generate $\vec{\Gamma}^{N,new}$ in the vicinity of $\vec{\Gamma}^{N,old}$).

(iii) Consider $\Delta E = E(\vec{\Gamma}^{N,new}) - E(\vec{\Gamma}^{N,old})$. If $\Delta E < 0$ (i.e., the energy goes down), then accept the new configuration $\vec{\Gamma}^{N,new}$. Otherwise, accept it with probability

$$\exp\left(-\Delta E/k_B T\right) \tag{3}$$

(iv) Repeat steps (ii) and (iii) many times while calculating averages of desired properties.

In some circumstances, the Metropolis algorithm does not represent the best method for simulating the physical system of interest. For example, in the simulation of some nonuniform systems, such as the solvent fluid contained between two very large colloidal particles, the appropriate ensemble in which to simulate the system is the so-called grand canonical ensemble (Adams, 1975; Snook and van Megen, 1981), in which the volume V, temperature T, and chemical potentials μ_i are fixed, and the numbers of molecules, N_i of species i, fluctuate and come to equilibrium during the course of the simulation. In this case, the vector in phase space, $\vec{\Gamma}$, is of variable length depending on the number of molecules in the simulation at any given point; it also depends on the species present in the case of a mixture. The algorithm employed in simulation is:

(i) Initialize the positions and number of the molecules (i.e., initialize $\vec{\Gamma}$).

(ii) Choose a molecule at random and move it randomly to a new position.

(iii) Add/delete a molecule of species i. [Steps (ii) and (iii) randomly generate $\vec{\Gamma}^{new}$ in the vicinity of $\vec{\Gamma}^{old}$.]

(iv) Accept $\vec{\Gamma}^{new}$ if E decreases; otherwise, accept it with probability

$$\exp(-\Delta E/k_B T) \tag{4}$$

for a move of type (ii). For moves involving addition or deletion of a species i molecule, the acceptance criterion depends on ΔE and the chemical potential of species i molecules (Adams, 1975).

(v) Repeat steps (ii) to (iv) many times.

This section is concluded with several observations about both the canonical ensemble and grand canonical ensemble Monte Carlo algorithms:

1. Asymptotically, the configurations generated will be low-energy configurations due to the nature of the probability distribution governing the ensembles.

2. The algorithms have the capability of 'climbing out' of local minima in the energy function since uphill moves are permissible.

3. The proportion of uphill moves accepted increases with temperature.

Physically, one type of local energy minimum that can arise is the amorphous glassy state which is obtained by a rapid temperature drop (*quench*) from a disordered fluid state. In fact, one can easily generate glassy state configurations using simulation by starting with a disordered liquid and rapidly quenching it. The likelihood of making a transition out of the quenched glassy state is extremely low, since to reach the lowest energy, ordered crystalline state an energy barrier must be overcome but the probability of making the necessary uphill moves is very low when the system temperature is low. Thus, in order to generate a lowest energy, crystalline state from an initially fluid system, one must lower the temperature very gradually, allowing the system to approach equilibrium at each temperature. This process of annealing, as opposed to quenching, is used to produce nonamorphous crystalline metals in real systems and provides the mechanism in a simulation for bringing a system into the lowest possible energy state.

Simulated annealing

Multivariable optimization is the general term used to describe the process of identifying the set of variables $\vec{x} = (x_1, x_2, \ldots, x_n)$ which globally minimize an objective function $C(\vec{x})$ (which, without loss of generality, can be regarded as the cost associated with the particular choice of $\vec{x}$). Kirkpatrick et al. (1983) conjectured that multivariable optimization problems involving a large number of variables might be solved efficiently using a generalization of the canonical ensemble Monte Carlo algorithm. An analogy was drawn between energy in canonical ensemble physical systems and cost in multivariable optimization and between the phase point $\vec{\Gamma}^N$ and the vector $\vec{x}$. Since, as noted above, if a physical system is annealed to extremely low temperature, it will eventually find itself in the lowest energy configuration, Kirkpatrick and coworkers suggested that by performing an analogous 'annealing' procedure on a multivariable optimization problem the vector $\vec{x}$ which globally minimized $C(\vec{x})$ would be located.

This identification requires that within the optimization context an analog of the temperature of physical systems must be identified. Kirkpatrick et al. introduced the concept of a simulated annealing temperature, T_{sa}, which has the units of cost and is used to control the probability for accepting uphill moves in cost. The simulated annealing temperature starts out at a high value so that a high proportion of attempted changes in $\vec{x}$ are accepted. As the simulation progresses, T_{sa} is reduced periodically, according to a prescription known as the *annealing schedule*. With each reduction in T_{sa}, the proportion of accepted moves goes down until, eventually, no moves are accepted. This implies that all attempted moves are uphill in cost and that the temperature is low enough that these moves are no longer accepted. Provided the annealing has been performed slowly enough, the final $\vec{x}$ should represent the global minimum of $C(\vec{x})$.

The simulated annealing algorithm of Kirkpatrick et al. (1983) is given by:

(i) Initialize $\vec{x}$ and T_{sa}.

(ii) Choose an element of $\vec{x}$ at random and change it randomly to a new value (i.e., randomly generate $\vec{x}^{new}$ in the vicinity of $\vec{x}^{old}$).

(iii) Consider $\Delta C = C(\vec{x}^{new}) - C(\vec{x}^{old})$. If $\Delta C < 0$ (i.e., the cost decreases), then accepted the new vector $\vec{x}^{new}$. Otherwise, accept it with probability.

$$\exp(-\Delta C/T_{sa}) \tag{5}$$

(iv) Repeat steps (ii) and (iii) many times while periodically reducing T_{sa}. The annealing schedule of Kirkpatrick et al. involved reducing the annealing temperature according to

$$T_{sa}^{(n+1)} = \xi T_{sa}^{(n)} \tag{6}$$

where ξ is a constant somewhat less than unity and n represents the nth annealing temperature. Notice that this algorithm is very similar to the canonical ensemble Metropolis algorithm for physical systems described above. The common feature that this algorithm and the canonical algorithm share, which differs from the grand canonical algorithm, is that the number of variables manipulated in each case is fixed; i.e., $\vec{\Gamma}^N$ and $\vec{x}$ have a constant number of elements. With this in mind, it is appropriate to categorize the simulated algorithm presented here as being appropriate for 'canonical' optimization problems.

As developed in this paper, there is a natural generalization of the grand canonical ensemble Monte Carlo simulation algorithm that would be applicable to 'grand canonical' optimization problems in which the number of variables to be manipulated is not fixed and is unknown before the optimization is performed.

In statistical mechanics, the critical element in the utility of Monte Carlo simulation is that the asymptotic distribution of points in phase space is independent of the starting configuration. This is proved using the mathematical theory of Markov chains (Hammersley and Handscomb, 1964) and suggests that in the optimization context, simulated annealing is capable of producing the global optimum independent of the initial guess. Several formal proofs have been given which establish that, if the number of attempted moves at each temperature is infinite, simulated annealing produces asymptotically the global optimum solution of combinatorial optimization problems with probability one (Romeo and Sangiovanni-Vincentelli, 1984; Lundy and Mees, 1984). In practice, one cannot guarantee that the solution obtained by simulated annealing in a finite amount of time is the rigorous optimum; however, the formal results suggest that a sufficiently slow annealing schedule will provide an optimal or near-optimal solution that is independent of the initial guess and therefore, in principle, avoids becoming trapped in local minima of the cost function that lie between the initial guess and the global optimum.

The random nature of the move used in a simulated annealing algorithm to create a new design might suggest that the method is similar to the adaptive random search algorithms described by Martin and Gaddy (1982). In the latter algorithms, to generate $\vec{x}^{new}$ and $\vec{x}^{old}$, many points $\vec{x}$ in the vicinity of $\vec{x}^{old}$ are generated randomly and the one with the lowest cost becomes $\vec{x}^{new}$. Thus, only downhill moves in cost are considered and the adaptive random search methods are therefore very likely to become trapped in a local minimum of the cost function. Consequently, the only similarity between simulated annealing and the adaptive random search method is that new designs are generated stochastically.

'Canonical' Optimization: Pipeline Networks

The general problem of optimal design of fluid transportation systems is common throughout the chemical processing industry. In many cases the objective is to minimize the transportation cost subject to regulatory and operating constraints. Normally, the layout of the transportation network is fixed, and the problem that remains to be solved is balancing pressure drop requirements against pumping cost or throughput.

Simulated annealing has been applied to two examples of fluid transportation network optimization: the design of a pressure relief header network (described in this section) and the design of a compressible fluid network (Wohlpart, 1987). Both problems are 'canonical' optimization problems since, in each case, the objective function (cost) is minimized with respect to a fixed number of variables (pipe diameters in the case of the pressure relief header network, pipe diameters and compressor duties in the case of the compressible fluid transport problem). The success of simulated annealing on these two problems serves to confirm that, with very little effort required on the part of the user, the algorithm is capable of obtaining a near minimum cost solution to canonical optimization problems that arise in chemical process design.

In the remainder of this section, the application of simulated annealing to pressure relief header network design will be described in detail. The role of a pressure relief header network in a chemical plant is to transport fluids as expediently as possible from process equipment to the flare head. This requires that sonic velocity be maintained in the discharge valves connecting the process equipment to the flare for an initial transitory period. Maintaining sonic velocity through a valve requires that the pressure on the downstream side of the valve remain below a certain value, which is determined by process conditions and the valve. This pressure sets a maximum allowable pressure drop between the valve and the flare. For a network, maximum allowable pressure drops are specified between downstream sides of valves and the flare, creating a set of constraints on the design.

We consider the problem of Cheng (1976), the qualitative features of which have been discussed by Cheng and Mah (1976). The problem has also been considered by Murtagh (1972), who describes a method that will give the optimum solution based on a continuous set of pipe diameters. Cheng solved the problem by the discrete merge method, which has its foundation in dynamic programming and is suitable for certain fixed structures. It can accommodate discrete pipe sizes and generates lists of pipe sections by serial and parallel merging.

Although Cheng and Mah (1976) describe the discrete merge method as applied to the pressure relief header network problem, neither the full problem description nor a solution is contained in their paper; Cheng (1976) gives the problem description, his computer program, and a solution. Figure 1 depicts the pressure relief network. The full problem description is given in Tables 1 and 2. As shown in Figure 1, there are 79 pipe segments and 34 source nodes. For each pipe segment there are nine possible choices, given at the bottom of Table 1, for the pipe segment diameter. It should be clear that the number of different combinations of pipe diameters is extremely large for this problem; in fact, with nine different pipe diameters possible for each segment, the number of possible designs is 9^{79}. Thus, any solution method employed must be selective of the solution space it searches.

The pressures at the upstream (A) and downstream (B) end of a pipe segment for compressible flow are related by

$$P_A^2 - P_B^2 = \frac{0.3697\, f l W^2 RT}{\pi^2 g_c M d^5}\left[1 + \frac{4.61\, d}{4fl}\log_{10}\left(\frac{P_A}{P_B}\right)\right] \quad (7)$$

where

$$l = \ell_s + \ell_{eq}$$

In this equation, ℓ_s is the straight length and ℓ_{eq} is the additional equivalent length. Following Cheng and Mah, the group in brackets in Eq. 7 (a kinetic energy correction term) is close to unity giving

$$P_A^2 - P_B^2 = \frac{K_i}{d^{4.814}} \quad (8)$$

In calculating temperatures of streams formed by mixing, the mixing rule used is simply

$$T = \sum_i W_i T_i \Big/ \sum_i W_i \quad (9)$$

where W_i is the mass flowrate of stream i.

The cost of the network is determined by the capital cost of the pipe segments. The cost of a pipe is assumed to be a linear function of its diameter. Thus, the cost of a pipe of diameter d is simply

$$C = (\alpha + \beta d)\, \ell_s \quad (10)$$

where α and β are constants. From Eq. 10, it is clear that the cost function in this optimization problem is linear in the variables $\vec{d} = (d_1, d_2, \ldots, d_N)$ where N is the number of pipe segments in the network. The nonlinearity in this problem arises through the pressure constraints which can be written as

$$b_i \geq \sum_{j \epsilon S_i} (P_{Aj}^2 - P_{Bj}^2) \quad (11)$$

where S_i is the set of pipes between node i and the flare. Values of b_i are listed in Table 1.

We developed our own computer program implementing the discrete merge method described by Rothfarb et al. (1970) and

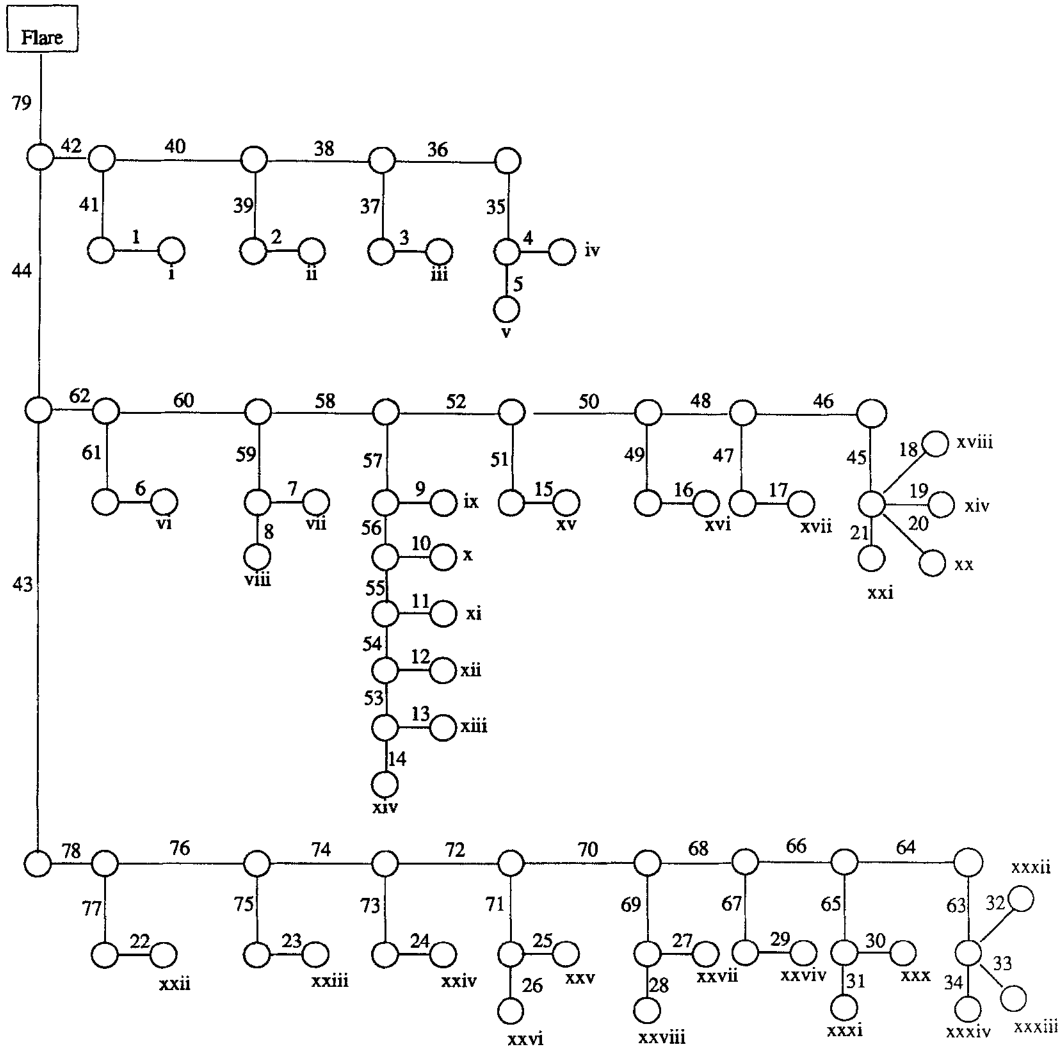

Figure 1. Pressure relief header network.

Table 1. Pressure Relief Header Network Node Properties

Node	*W* kg/s	*T* K	*M*	$b_i \times 10^{-12}$ Pa^2	Node	*F* kg/s	*T* K	*M*	$b_i \times 10^{-12}$ Pa^2
i	0.126	222.2	4.0	3.3277	xviii	0.378	222.2	4.0	3.8109
ii	2.520	222.2	4.0	3.3844	xix	0.126	222.2	4.0	3.8109
iii	2.520	555.6	10.0	3.3844	xx	0.126	555.6	10.0	3.8109
iv	0.630	277.8	5.0	3.3844	xxi	0.126	277.8	5.0	3.9247
v	1.260	444.4	8.0	3.4507	xxii	1.260	444.4	8.0	4.0480
vi	0.252	1,111.1	10.0	3.4507	xxiii	1.008	222.2	4.0	4.1808
vii	0.252	222.2	4.0	3.4507	xxiv	1.260	555.6	10.0	4.4750
viii	0.252	222.2	4.0	3.4507	xxv	0.378	277.8	5.0	4.4750
ix	0.252	555.6	10.0	3.5265	xxvi	0.378	444.4	8.0	4.6363
x	0.252	277.8	5.0	3.5265	xxvii	0.126	1,111.1	10.0	4.8072
xi	0.126	444.4	8.0	3.6118	xxviii	0.630	222.2	4.0	4.3231
xii	0.252	1,111.1	10.0	3.6118	xxix	1.890	222.2	4.0	4.8072
xiii	0.882	222.2	4.0	3.6118	xxx	1.260	222.2	4.0	5.0351
xiv	1.008	222.2	4.0	3.7066	xxxi	1.260	1,111.1	10.0	4.1808
xv	3.780	555.6	10.0	3.7066	xxxii	0.756	555.6	10.0	5.1774
xvi	3.780	277.8	5.0	3.7066	xxxiii	0.252	555.6	10.0	5.1774
xvii	0.378	444.4	8.0	3.7066	xxxiv	0.504	555.6	10.0	5.1774
Pipe Index	1	2	3	4	5	6	7	8	9
d, m	0.0508	0.1016	0.1524	0.2032	0.2540	0.3048	0.3556	0.4064	0.4572

Table 2. Pressure Relief Header Network Lengths

Branch	l_s, m	l_{eq}, m	Branch	l_s, m	l_{eq}, m
1	4.57	9.14	41	41.15	41.15
2	15.54	12.19	42	44.20	33.53
3	15.54	12.19	43	45.42	67.06
4	3.96	12.19	44	1.52	0.30
5	13.72	12.19	45	5.49	6.10
6	2.13	12.19	46	6.10	30.48
7	8.53	30.48	47	4.88	24.38
8	9.45	30.48	48	26.52	42.67
9	4.88	30.48	49	20.42	12.19
10	18.29	36.58	50	4.57	30.48
11	10.67	48.77	51	24.38	6.10
12	9.45	42.67	52	9.45	36.58
13	4.57	24.38	53	41.45	6.10
14	42.67	30.48	54	15.24	6.10
15	3.05	36.58	55	33.22	79.55
16	3.05	24.38	56	10.97	24.38
17	0.30	0.30	57	4.27	9.14
18	3.05	24.38	58	4.57	6.10
19	3.05	24.38	59	13.11	9.14
20	3.05	24.38	60	2.44	6.10
21	3.05	24.38	61	8.53	18.29
22	0.30	0.30	62	0.30	0.30
23	0.30	0.30	63	9.14	6.10
24	0.30	0.30	64	11.58	24.38
25	1.22	12.19	65	0.61	6.10
26	1.22	18.29	66	7.01	24.38
27	3.05	36.58	67	7.92	6.10
28	3.05	18.29	68	17.68	24.38
29	2.44	18.29	69	3.05	6.10
30	3.35	18.29	70	5.49	24.38
31	3.35	24.38	71	3.96	6.10
32	25.30	36.58	72	7.01	24.38
33	23.16	30.48	73	0.61	6.10
34	24.38	33.53	74	7.01	24.38
35	14.63	9.14	75	1.22	6.10
36	10.67	6.10	76	3.96	24.38
37	12.80	9.14	77	21.34	18.29
38	19.81	6.10	78	0.61	0.30
39	20.73	6.10	79	1,005.84	335.28
40	20.73	9.14	—	—	—

referred to by Cheng (1976). The earlier paper, which introduced the discrete merge method for optimizing gas transportation networks, did not employ list shortening techniques introduced into the method by Cheng and Mah. List shortening, while reducing memory requirements, can lead to suboptimal solutions. We reduced memory requirements in our implementation of the method by dynamically allocating array space as needed using the C programming language. All computations reported in this paper were performed on a Sun 3/50 engineering workstation without a floating point accelerator board. The discrete merge solution required 210 seconds of cpu time. Table 3 gives the pipe diameter indices for the various branches. Note that very few large pipes are present. Our network cost of \$165,075 by the discrete merge method is considerably less than the \$200,851 cost reported by Cheng (1976).

Applying the canonical form of simulated annealing described above to the design of pressure relief headers is straightforward. We used a single move which works as follows. A random number is used to select a pipe segment. A second random number determines whether an attempt is made to increase or decrease the diameter of the pipe segment by *one* discrete pipe size. Constraints are checked and if satisfied the change in cost of the network is calculated; note that only a single pipe segment changes cost. We accept or reject the move according to the acceptance criteria described earlier. Initially, all pipe diameters in the pressure relief header network are set to the largest possible diameter; this is the feasible design furthest from the cost optimum. The annealing schedule involves attempting a fixed number of moves N_m at each annealing temperature, then decreasing the annealing temperature to a fraction ξ of its current value, as given by Eq. 6, and continuing. The initial T_{sa} was \$10,000. When T_{sa} became less than unity and no attempted moves at the current T_{sa} were accepted, the algorithm was terminated.

In checking the pressure drop constraints, first the change in pressure drop across the chosen pipe is calculated. Then, a list is generated of the pressure drop constraints that are affected by the chosen pipe. The change in pressure drop is added to the prior total pressure drops along the affected paths. It is then pos-

Table 3. Discrete Merge Solution to Pressure Relief Header Network

Branch	Pipe	Branch	Pipe	Branch	Pipe	Branch	Pipe
1	1	21	1	41	1	61	1
2	2	22	1	42	3	62	4
3	2	23	1	43	4	63	2
4	1	24	1	44	4	64	2
5	2	25	1	45	1	65	2
6	1	26	1	46	2	66	3
7	1	27	1	47	1	67	2
8	1	28	1	48	2	68	3
9	1	29	2	49	2	69	1
10	1	30	1	50	3	70	4
11	1	31	2	51	2	71	1
12	1	32	1	52	3	72	3
13	2	33	1	53	2	73	1
14	2	34	1	54	2	74	3
15	2	35	2	55	2	75	1
16	2	36	2	56	3	76	4
17	1	37	2	57	2	77	2
18	1	38	3	58	4	78	3
19	1	39	2	59	1	79	7
20	1	40	3	60	4	—	—

sible to determine whether all pressures are within their respective limits. If all constraints are not satisfied, the move is rejected at that point.

According to Aarts and Laarhoven (1985), in a simulated annealing algorithm the number of moves attempted at each T_{sa} need only be as large as the number of states reachable from any given state. In the present case, this is $N_m = 158$, which we used in our calculations.

Several interesting features were observed in applying simulated annealing to the pressure relief header network. First, computing times were found to be strongly dependent on the choice of random number generator; an efficient, simple random number generator gave a fifty-fold decrease in computing time compared to a system random number generator. Second, for $\xi \leq 0.5$, simulated annealing is close to a random search method. This is shown in Table 4 which gives the effect of ξ on the mean and standard deviation of the final cost for ten different seeds to the random number generator. For $\xi \leq 0.5$, T_{sa} is reduced so quickly that few uphill moves in cost are accepted, with the result that the algorithm becomes trapped in a high-cost local minimum. Solutions obtained for $\xi > 0.5$ differ significantly from those obtained for $\xi \leq 0.5$. Third, for $\xi > 0.7$, solutions are within 0.4% of the global minimum. At $\xi = 0.7$, only six seconds of computing time are required on the Sun 3/50.

With $\xi = 0.99$, simulated annealing found the global minimum (Table 3) for 20% of the initial seeds to the random number generator. When the global minimum was not found, a solution very close to it was obtained. Interestingly, with $\xi = 0.99$, simulated annealing explores only 10^{-71} of the full solution space of 9^{79} combinations. The computer time required for simulated annealing with $\xi = 0.99$ and an efficient random number generator is 180 seconds, equivalent in order to that required by the discrete merge method.

It should be emphasized that the advantage of simulated annealing is its ability to easily adapt to more complex problems for which discrete merge is not applicable. For example, if one considers the more general problem of optimizing the layout, connectivity and sizing of the pressure relief network (instead of simply focusing on the sizing of a network whose layout and connectivity is already specified), simulated annealing is able to handle the more general design problem with little modification to the annealing algorithm. In contrast the discrete merge technique would not be applicable to the more general problem.

'Grand Canonical' Optimization: Heat Exchanger Networks

The requirement for greater heat integration in chemical processes has led to a concerted effort to develop methods for synthesizing optimal heat exchanger networks (HEN's). Optimality in HEN's has been associated with two features: maximum energy recovery (MER), which means that the HEN recovers all heating and cooling duty available in the process streams; and minimum number of heat exchanger units, a consequence of the often assumed nonlinear dependence of the cost of heat exchangers of the exchanger area, which implies that two exchangers of a given total area will cost more than a single exchanger of the same area. So ubiquitous are these two concepts that methods for identifying MER targets and minimum exchanger number (MEN) have become the central focus of many research efforts, the synthesis of a network to meet these targets being regarded as a relatively straightforward exercise. Design algorithms are routinely evaluated according to whether they predict MER and/or MEN.

Table 4. Effect of ξ on Average Final Cost for $N_m = 158$

ξ	$\bar{C}$, \$	σ_C, \$
0.10	170,077	4,998
0.20	170,232	5,249
0.30	170,213	4,761
0.40	170,811	4,824
0.50	170,427	5,188
0.60	167,293	1,452
0.70	166,600	586
0.80	165,591	253
0.90	165,619	494
0.95	165,169	78

Linnhoff and Flower (1978) were among the first authors to recognize that it was indeed possible to predict MER and MEN. Subsequently, Linnhoff and Hindmarsh (1983) developed the pinch design technique. The pinch design method enables MER and MEN to be predicted from the stream data and prescribes a method for synthesizing the HEN by designing away from the pinch. Cerda et al. (1983) pursued a different approach to this problem by casting it into the form of the transportation model of operations research. In this model, the problem again is broken into temperature intervals as was done by Linnhoff and Flower (1978). The resulting problem of finding MER can then be computed by shipping the hot streams' heat to the cold streams. (A hot/cold stream is a stream whose final temperature is lower/higher than its initial temperature.) The resulting transportation problem can be easily solved by linear programming.

Papoulias and Grossmann (1983) solved the HEN problem by using a transhipment model of operations research. In this model the hot streams and cold streams in a given temperature interval are shipped to intermediate warehouses where they mix. Any residual heat is then sent to the next lowest temperature interval. In order to find the structure, the streams are matched in one or more intervals so as to produce a minimum number of matches. Floudas et al. (1986) extended the earlier work of Papoulias and Grossmann by developing a method of finding the optimum distribution of areas once the structural layout of the network was determined and by insuring the existence of a one-to-one correspondence between the matches in the temperature intervals and actual units.

These design algorithms all assume that an optimal HEN is characterized by MER and the MEN consistent with MER. However, it should be noted that this is only a heuristic that follows from assuming that operating costs are dominated by energy consumption and that capital cost is a non-linear function of exchanger area. In practice, it is not clear that this heuristic will always characterize the lowest cost HEN. For example, a trade-off between capital cost and utilities may indicate lower overall cost for a HEN which does not achieve MER, but has fewer heat exchangers than an MER network.

It is clear that the role of heuristics in HEN synthesis is to limit the size of the optimization problem since, in their absence, the problem is combinatorially large. From the discussion above, combinatorially large problems are well suited to solution by simulated annealing. Thus, in applying simulated annealing to HEN synthesis, all heuristics are discarded and a global cost function is minimized with respect to the design variables.

The simulated annealing algorithm required for HEN synthesis is based on analogy with the Metropolis algorithm for simulating the grand canonical ensemble. The grand canonical simulated annealing (GCSA) algorithm for the design of a network of heat exchangers allows the number of exchangers to change as the design evolves. To the authors' knowledge, our implementation of the simulated annealing algorithm for heat exchanger network design is the first time that simulated annealing has been applied to any multivariable optimization problem in which the number of variables is not fixed.

The successful application of GCSA to two previously published literature problems was described by the authors in an earlier publication (Dolan et al., 1987). In one case, we found a lower cost design to a problem posed and solved by Linnhoff and Turner (1981) using pinch technology. In the other case, for the infeasible match 4SP1 problem introduced by Papoulias and Grossmann (1983), we found a design costing 35% less than that proposed by Papoulias and Grossmann. In the infeasible match problem, heat exchange is not permitted between two or more streams for safety, plant layout, or operational reasons. Our solution involved the concept of hot-to-hot and cold-to-cold matches (heat exchange between streams of the same type). Such solutions were apparently first proposed on heuristic grounds by Grimes et al. (1982) and have since been found algorithmically by us and by Viswanathan and Evans (1987), who refer to such matches as dual stream solutions, since a hot or cold stream plays a role of both a hot and a cold stream in different parts of the network.

Several issues concerning algorithms and data structures have been resolved in efficiently implementing GCSA for HEN synthesis. A full description is beyond the scope of the present paper and will be discussed in a future publication; in this paper, we mention the major innovations only briefly. The essential feature of the implementation is that the structure of the network is represented by a multiply-linked list in which heat exchangers, splitters, and combiners in the network are represented by records. The streams connecting these unit operations in the network are represented by pointers. Any heating or cooling in the network which is not satisfied by process-to-process heat exchange is automatically provided by utilities. Use of the linked list data structure makes it possible to calculate the difference in cost between two network designs, ΔC, without calculating the cost of the two separate networks. Being able to calculate ΔC directly yields an algorithm which is two orders of magnitude faster than an algorithm that costs each network design separately. The linked list structure also permits the efficient addition and deletion of exchangers and splits to the network. The program was written in C and runs interactively on the Sun workstation with an icon-driven user interface.

The moves play an extremely important role in HEN design. The moves chosen in the present study allow the simulation to change topological structures relatively easily by allowing any of the energy transferred at a given match to be obtained from the residual energy in a stream (i.e., the current utility duty of the stream) or from a combination of the residual energy in a stream and energy from another unit. Thus, it is possible to stay at the current level of energy recovery at any given point in the simulation and still vary the topology. This permits the implementation of relatively rapid annealing schedules since fewer uphill moves in cost are required to change structure. After some experimentation, the following moves were developed and found to yield an efficient search of the feasible solution space. At each iteration of the GCSA algorithm, one of these moves is performed; each move is equally probable.

1. Add a heat exchanger at a randomly chosen point in the network. The heat duty q for this heat exchanger is set either randomly (with the maximum q being the smaller of the current utility demands of the two streams) or in combination with shifting q from another exchanger in the network (with the maximum shifted q being 20% of the heat duty on the other exchanger).
2. Delete a randomly chosen heat exchanger. If a split was involved, then eliminate the split if a side stream has no heat exchangers.
3. Add or delete a random amount of q from a randomly chosen heat exchanger. The maximum amount of q added or deleted is 20% of the current q.
4. Shift a random amount of q from one randomly chosen heat exchanger to another randomly chosen heat exchanger. The maximum amount of q shifted is 40% of the minimum of the two heat loads.
5. Split a randomly chosen stream and add a heat exchanger to the side stream. The q is chosen as in move 1.
6. Change the split ratio of a randomly chosen splitter by adding a random fraction between -0.2 and 0.2. Split ratios greater than unity and less than zero are set to unity and zero, respectively.

Any move which violates imposed constraints (such as an infeasible match or a prescribed minimum approach temperature ΔT_{min}) is rejected. After each move, energy balances (and material balances when necessary) are solved efficiently through traversal of the linked list data structure used to represent the network. As in the earlier study of GCSA (Dolan et al., 1987), the initial design for each network consists of no heat exchangers with all heating and cooling requirements being met by utilities. We remark that experience with many HEN synthesis problems has demonstrated that the execution time per move (including evaluation of ΔC) in the GCSA algorithm is essentially independent of problem size. This feature of our algorithm is shared by the most efficient implementations of simulated annealing for the traveling salesman problem (Press et al. 1986; Aarts and van Laarhoven, 1985).

In this paper, we apply GCSA to three heat exchanger synthesis problems. The first is the 7SP4 problem considered by Floudas et al. (1986) using mixed integer nonlinear programming. Details of the problem description and the cost function are provided by Floudas et al. and need not be repeated here. Floudas et al. considered a global cost minimization problem in which the minimum approach temperature ΔT_{min} was a variable determined as part of the optimization procedure. In GCSA, we simply set $\Delta T_{min} = 0$.

We adopted the annealing schedule of Aarts and van Laarhoven (1985), which is based on maintaining quasi-equilibrium at each temperature. This schedule is given by

$$T_{sa}^{new} = T_{sa}^{old}\left\{1 + \frac{\ln(1+\gamma)T_{sa}^{old}}{3\sigma}\right\}^{-1} \tag{12}$$

where σ is the standard deviation of the cost at the current annealing temperature T_{sa}^{old} and γ is a parameter which governs the speed of the annealing. The length of the annealing schedule increases with decreasing γ. In the present study, we set γ =

0.01 and used an initial T_{sa} of $55,000 and 400 attempted moves at each temperature. The GCSA algorithm was terminated when $T_{sa} \leq 1$. [Alternative annealing schedules that minimize the number of moves at each simulated annealing temperature are described by Kirkpatrick et al. (1983) and Romeo and Sangiovanni-Vincentelli (1984). Based on these, our value of γ results in a very conservative (i.e., slow) annealing schedule.]

The solution obtained by GCSA to the 7SP4 problem of Floudas et al. (1986) is shown in Figure 2. There are six hot streams and one cold stream in the problem. The cold stream flows from left to right and hot streams flow from right to left. Heat exchangers are shown connecting the streams. We have the same number of units as the design of Floudas et al. However, our design has the unusual feature of placing an exchanger (connecting stream 2 and one of the splits on stream 1) across the pinch. Using the cost function of Floudas et al., our network has an annualized cost of $644,480. We calculate the cost of the network of Floudas et al. to be $645,695. Thus, the result obtained by GCSA is slightly lower in cost. (Note that there is an error in the exchanger areas of the final design in the original paper of Floudas et al. which was corrected in errata (see literature cited). The $645,695 cost of the final design quoted above is based on the corrected areas and is lower than the cost of $647,050 reported in the original paper.) The design shown in Figure 2 required approximately 400 minutes of cpu time on the Sun 3/50 workstation. This appears to be about one order of magnitude more computer time than that required by Floudas et al. (A direct comparison is difficult due to differences between the two computers used.)

In Figure 3, the acceptance rate of each of the GCSA moves enumerated above and the total cost of the network are shown as a function of T_{sa}. Each point on the graph corresponds to an average over many annealing temperatures. The structural moves (1, 2 and 5) which alter the topology of the network are lower in acceptance probability than the parametric moves (3, 4 and 6) because structural moves result in larger changes in the cost of the network. As the temperature is lowered, the acceptance probability of structural moves becomes very small and over the final one-third of the values of T_{sa} (not shown in the figure), for the most part, only parametric alterations to the network are accepted.

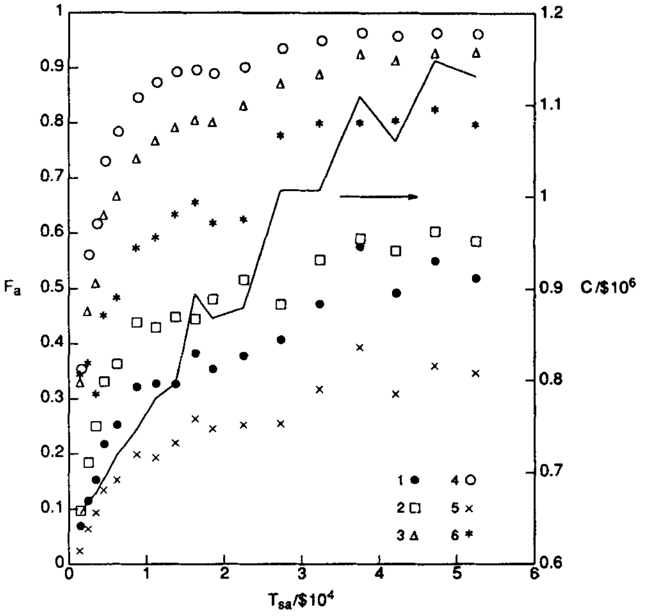

Figure 3. Acceptance rate of the six moves in the GCSA algorithm for the 7SP4 problem.

As a second example, we consider the 7SP4 problem with a discontinuous cost function. As mentioned previously, off-the-shelf heat exchangers cost between 50% and 75% of custom fabricated units (Rubin et al., 1984). For our purposes, assume that

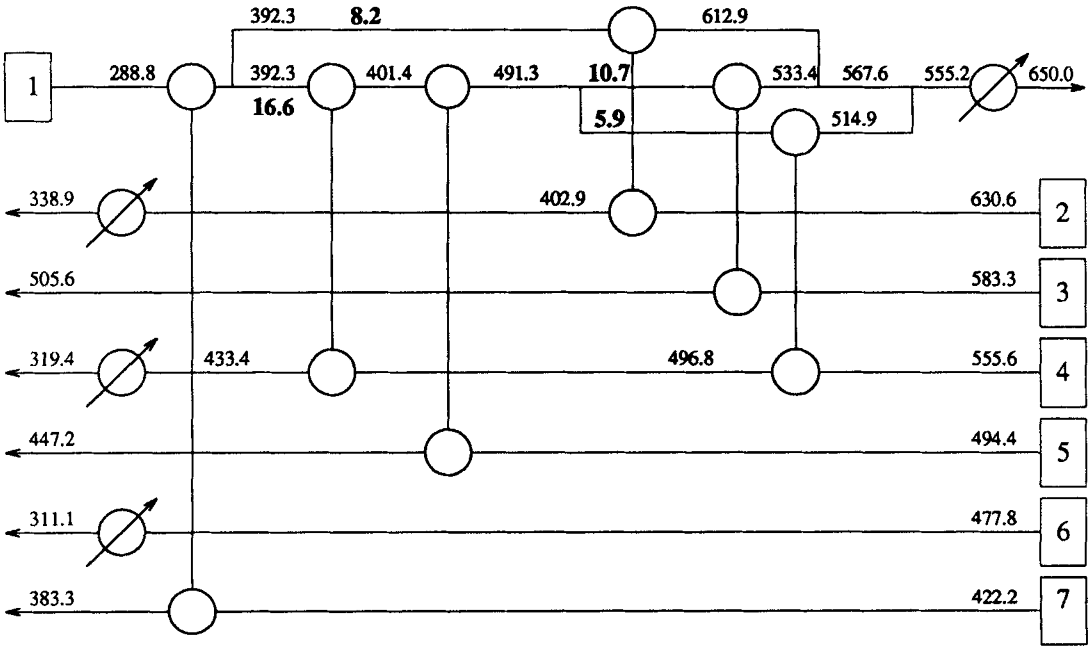

Figure 2. Solution of 7SP4 problem obtained by simulated annealing.

Flow rates on splits are shown in boldface.

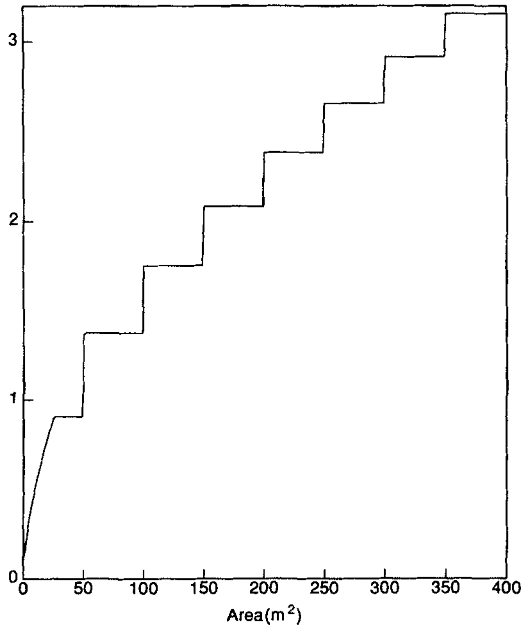

Figure 4. Cost of heat exchangers as a function of area for the cost function defined in Eq. 13.

off the shelf exchangers are available in increments of 50 m^2 and that the cost of bypasses is insignificant. For this case, a suitable cost function for an exchanger of area A is given by

$$C_{\text{actual}} = \min\left\{C_{FCG}(A), \frac{2}{3}\,C_{FCG}(A')\right\} \tag{13}$$

where

$$A' = 50[(A/50) + 1]$$

In these equations, C_{actual} is the cost of an exchanger, C_{FCG} is the cost based on the cost function of Floudas et al. (1986), and $[x]$ is the integer part of x. In practical terms, Eq. 13 indicates that a larger off-the-shelf exchanger should be used if it would cost less than a custom fabricated one. This results in a cost function which has discontinuities as shown in Figure 4. The GCSA algorithm is not affected by this choice of cost function since it is derivative-free and involves no assumption about monotonicity.

The network design for 7SP4 obtained by the GCSA algorithm using Eq. 13 as the cost function is shown in Figure 5. It has an annualized cost of $601,041. For comparison, the cost of the Floudas et al. design using this cost function is $610,343 and the cost of the design shown in Figure 2 is $606,927. The design obtained by GCSA for the discontinuous cost function has eleven units (one more than the minimum number for this network) and exhibits cross pinch heat transfer. It is evident that the cost function of Eq. 13 will not, in general, lead to a MEN network. This is because with this cost function, for exchangers which are a little larger than a multiple of 50 m^2, it is cheaper to use an off-the-shelf exchanger with a small custom made exchanger than to use an off-the-shelf exchanger of the next larger size. This is illustrated by the two small utility exchangers on streams 3 and 7.

For the third example of heat exchanger network synthesis by GCSA, we consider a prohibited match problem that was introduced and studied by Linnhoff and Flower (1978), who also proposed a cost function for the problem. It is the 5SP1 problem with a match prohibited between streams 1 and 4. The design of

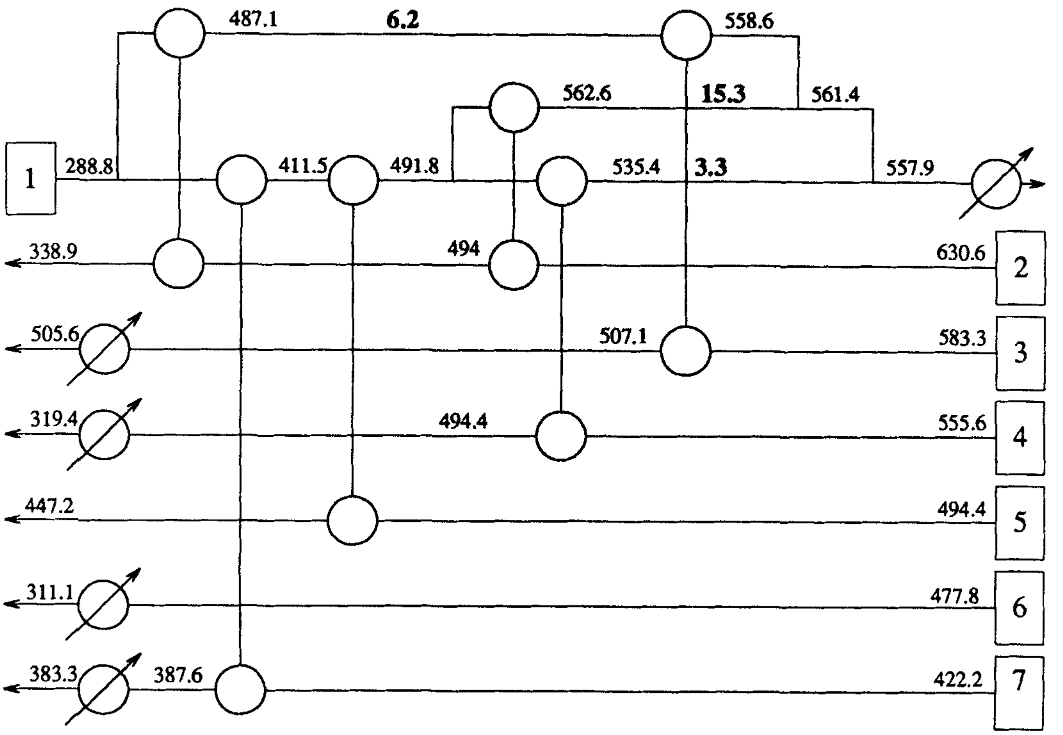

Figure 5. Solution of 7SP4 problem obtained by simulated annealing with cost function defined in Eq. 13.
Flow rates on splits are shown in boldface. All heat exchangers have bypasses.

Linnhoff and Flower (1978) had an annualized cost of \$50,340. Grimes et al. (1982) found a solution costing \$39,706 per year that featured a cold-to-cold match. In Figure 6, we show a design obtained by GCSA for the problem. The network structure is similar to that of Grimes et al. and has a cold-to-cold match between streams 1 and 3. This network has an annualized cost of \$39,440. Like the design of Grimes et al., our design achieves MER (while the design of Linnhoff and Flower does not) and contains one fewer unit that the Linnhoff and Flower design.

The method proposed by Cerda et al. (1983) to reach MER when a match is forbidden between two given streams requires the introduction of an intermediate stream, the purpose of which is to circulate a fluid in a closed loop between the two streams. Capital investment is clearly greater using this approach since a single exchanger in the constraint-free MER design is replaced by two exchangers and the additional piping necessary for the intermediate stream. By using hot-to-hot and cold-to-cold matches, the designs found algorithmically for forbidden match problems by GCSA and by the method of Viswanathan and Evans (1987) circumvent the need for intermediate streams, essentially by using existing process streams for the same purpose.

Conclusions

Simulated annealing, a new design technique for finding optimal and near-optimal solutions to process network design problems that have combinatorially large numbers of feasible designs, has been described and illustrated with examples. Problems have been classified as 'canonical' or 'grand canonical' depending upon whether or not the number of elements in the problem is fixed. The concept of the grand canonical simulated annealing (GCSA) problem, introduced here for heat exchanger networks, is new.

Simulated annealing does not rely on heuristics to find minimum cost solutions and is able to handle a wide variety of problems with no restrictions on the nature of the cost function. It is particularly well suited to optimization problems involving many variables which suggests that it is applicable to industrial-scale design problems.

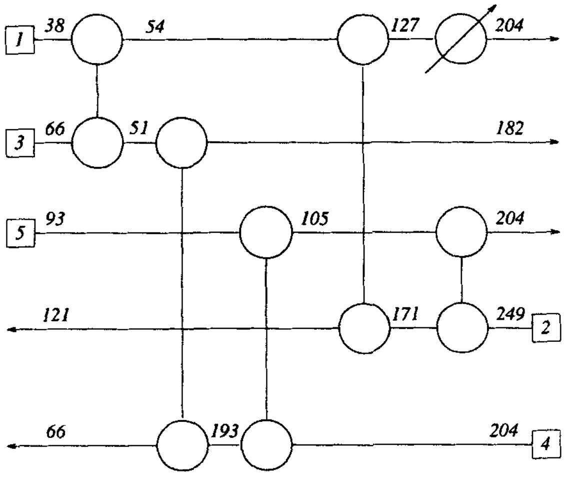

Figure 6. Solution of 5SP1 problem with prohibited match obtained by simulated annealing.
A match is prohibited between streams 1 and 4.

Acknowledgment

The authors are grateful to the Center for Computer-Aided Engineering and the Academic Computing Center at the University of Virginia for the provision of computational resources. The authors are indebted to R. S. H. Mah for correspondence on the pressure-relief header network problem and for providing the authors with a copy of W. B. Cheng's M.S. thesis.

Notation

C = cost, \$
d = diameter of a pipe, m
f = friction factor, 0.0475 (2.8186 × 10^6 $W/\mu d)^{-0.186}$
F_a = fraction of moves accepted (Figure 3)
g_c = gravitational constant, $kg \cdot m/(Ns^2)$
K_i = back pressure constraint constant (Eq. 11)
ℓ_{eq} = equivalent length, m
ℓ_s = straight length, m
M = molecular weight, $kg/kmol$
N_m = number of moves attempted at each simulated annealing temperature
P_A = upstream pressure, Pa
P_B = downstream pressure, Pa
q = heat transfer rate, kJ/s
R = gas constant
T = absolute temperature, K
T_{sa} = simulated annealing temperature, \$
W = mass flowrate, kg/s

Greek letters

α = cost coefficient, 7.1243 $\$/m$
β = cost coefficient, 341.05 $\$/m^2$
γ = parameter governing annealing rate (Eq. 12)
μ = viscosity, 2 × 10^{-5} $N \cdot s/m^2$
σ = standard deviation (Table 4)
ξ = parameter governing annealing rate

Literature Cited

Aarts, E. H. L., and P. J. M. van Laarhoven, "Statistical Cooling: A General Approach to Combinatorial Optimization Problems," *Philips J. Res.*, **40**, 193 (1985).

Adams, D. J., "Grand Canonical Ensemble Monte-Carlo for a Lennard-Jones Fluid," *Molec. Phys.*, **29**, 307 (1975).

Cheng, W. B., "Computational Techniques for Simple Pipeline Networks Design," M. S. Thesis, Dept. of Chemical Engineering, Northwestern Univ. (1976).

Cheng, W., and R. S. H. Mah, "Optimal Design of Pressure-Relieving Networks by Discrete Merging," *AIChE J.*, **22**, 471 (1976).

Cerda, J., A. W. Westerberg, D. R. Mason, and B. Linnhoff, "Minimum Utility Usage of Constrained Heat Exchanger Synthesis—A Transportation Problem," *Chem. Eng. Sci.*, **38**, 373 (1983).

Dolan, W. B., P. T. Cummings, and M. D. LeVan, "Heat Exchanger Network Design by Simulated Annealing," G. V. Reklaitis, and H. D. Spriggs, eds., *Foundations of Computer-Aided Process Operations*, CACHE, Elsevier, New York, 639 (1987).

Floudas, C. A., A. R. Circ, and I. E. Grossmann, "Automatic Synthesis of Optimum Heat Exchanger Configurations," *AIChE J.*, **32**, 276 (1986), errata, **32**, 1406 (1986).

Grimes, L. E., M. D. Rychener, and A. W. Westerberg, "The Synthesis and Evolution of Networks of Heat Exchange that Feature the Minimum Number of Units," *Chem. Eng. Commun.*, **14**, 339 (1982).

Hammersley, J. M., and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London (1964).

Karlin, S., *A First Course in Stochastic Processes*, Academic Press, New York (1973).

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Sci.*, **220**, 671 (1983).

Linnhoff, B., and J. R. Flower, "Synthesis of Heat Exchange Networks: I. Systemic Generation of Energy Optimal Networks," *AIChE J.*, **24**, 642 (1978).

Linnhoff, B., and E. Hindmarsh, "The Pinch Design Method for Heat Exchanger Networks," *Chem. Eng. Sci.*, **38**, 745 (1983).

Linnhoff, B., and J. A. Turner, "Heat Exchanger Networks: New Insights Yield Big Savings," *Chem. Eng.*, **88**(22), 56 (1981).

Lundy, M., and A. Mees, "Convergence of the Annealing Algorithm," *Simulated Annealing Workshop*, Yorktown Heights (Apr., 1984).

Martin, D. L., and J. L. Gaddy, "Process Optimization with the Adaptive Randomly Directed Search," *AIChE Symp. Ser.*, **78**(214), 99 (1982).

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, **21**, 1087 (1953).

Murtagh, B. A., "An Approach to the Optimal Design of Networks," *Chem. Eng. Sci.*, **27**, 1131 (1972).

Papoulias, S. A., and I. E. Grossmann, "A Structural Optimization Approach in Process Synthesis: II. Heat Recovery Networks," *Comp. Chem. Eng.*, **7**, 707 (1983).

Press, W. H., B. P. Flannery, W. T. Teukolsky and W. T. Vetterling, "Numerical Recipes: the Art of Scientific Programming," Cambridge University Press, Cambridge (1986).

Romeo, F., and A. Sangiovanni-Vincentelli, "Probabilistic Hill Climbing Algorithms: Properties and Applications," ERL, College of Engineering, Univ. of California, Berkeley (1984).

Rubin, F. L., H. A. Moak, A. D. Holt, F. C. Standiford, and D. Stuhlbarg, "Heat-Transfer Equipment" R. H. Perry and D. Green, eds., *Perry's Chemical Engineers' Handbook*, McGraw-Hill, New York, 11 (1964).

Rothfarb, B., H. Frank, D. M. Rosenbaum, K. Steiglitz, and D. J. Kleitman, "Optimal Design of Offshore Natural-Gas Pipeline Systems," *Oper. Res.*, **18**, 992 (1970).

Snook, I. K., and W. van Megen, "Calculation of Solvation Forces between Solid Particles Immersed in a Simple Fluid," *J. Chem. Soc. Farad. Trans. II*, **77**, 181 (1981).

Valleau, J. P., and S. G. Whittington, "A Guide to Monte Carlo for Statistical Mechanics: I. Highways," B. J. Berne, ed., *Modern Theoretical Chemistry: 5. Statistical Mechanics*, Plenum Press, New York, 134 (1977).

Viswanathan, M., and Evans, L. B., "Studies in the Heat Integration of Chemical Process Plants," *AIChE J.*, **33**, 1781 (1987).

Wohlpart, D., "Design of Pipe Networks by Simulated Annealing," Univ. of Virginia, unpublished (1987).

Manuscript received Aug. 2, 1988, and revision received Dec. 29, 1988.