Pergamon

Computers chem. Engng Vol. 18, No. 5, pp. 369–381, 1994
Copyright © 1994 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0098-1354/94 $6.00+0.00

SYNTHESIZING OPTIMAL DESIGN CONFIGURATIONS FOR A BRAYTON CYCLE POWER PLANT

L. A. Painton and U. M. Diwekar

Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

(Received 21 December 1992; final revision received 28 June 1993; received for publication 27 September 1993)

Abstract—This paper demonstrates the use of the simulated annealing method for the analysis of design configurations for a Brayton cycle power plant. This application centers aound a sequential modular chemical process simulator, the Space Nuclear Auxiliary Power System (SNAPS), designed to simulate the operation of power plants with different energy conversion techniques, to find optimal characteristics constrained by feasibility limitations, and to design the plant for maximum flexibility and reliability. The modular nature of SNAPS allows for the analysis of numerous configurations of nuclear power plants. Since the solution space is characterized by a large number of design and structural parameter combinations, it demonstrates the successful applicability of simulated annealing as a synthesis tool for nuclear power plant design problems. Simulated annealing is found to perform very well in the face of a large number of discontinuities in the solution space.

INTRODUCTION

Sequential modular chemical process simulators, such as ASPEN, have been extensively used for design of new processes and analysis of existing processes. These flowsheeting programs contain detailed models for calculating mass and energy balances across process units, as well as for sizing and costing (Diwekar *et al.*, 1992). SNAPS (The Space Nuclear Auxiliary Power System) is a chemical process simulator tailored specifically for space nuclear power plants (Diwekar and Morel, 1992). Based on the concept of sequential modular simulators, SNAPS can be used to simulate different configurations of space nuclear power plants or any power plant in general. In power plants energy balance plays an important role because in most cases the stream compositions or flowrates remain the same throughout the plant. This is unlike chemical plants where both material and energy balances are to be considered. SNAPS is specifically designed for power plants where the point concept, first proposed by Hughes *et al.* (1981), is used for storing the property vector and the stream vector contains the molar flowrates which remain constant for most of the configurations. SNAPS allows for simulation of numerous possible configurations of power plants. This flexibility of SNAPS, although very useful in analysis, poses the challenging synthesis problem of selecting optimal configurations from the various configuration sets to achieve specific goals.

The "synthesis" problem in design is to select the optimal design configuration, both in terms of the optimal structural topology and the optimal parameter level settings for a system to meet specified performance and cost objectives (Grossmann, 1990). This problem is very important for many practical applications and has gained much attention in recent years. One of the most important goals in chemical process synthesis is to create flowsheet structures.

Past approaches to the process synthesis problem have fallen in two areas:

(1) heuristics;
(2) mathematical programming techniques.

Heuristic approaches exploit physical intuition and engineering design knowledge to simplify the problem and quickly identify solutions which may be of good quality (Douglas, 1988). However, they do not guarantee optimality and do not account for tradeoffs and interactions very well.

The mathematical programming appoach to process synthesis involves: (a) formulation of a flowsheet superstructure incorporating all the alternative process configurations; and (b) identification of both the optimal configuration and operating process parameters by an algorithm based on the alternating sequence of nonlinear programs (NLPs) and mixed integer linear programs (MILPs). Over the last few years, significant advances in the mixed integer nonlinear programming (MINLP) algorithms have led to rapid developments of equation oriented software packages such as APROS (Paules and Floudas, 1989), DICOPT++ (Kocis and Grossmann, 1989; Viswanathan and Grossmann, 1990) and PROSYN

(Kravanja and Grossmann, 1990). Although these packages provide an environment for solving MINLP process synthesis problems, they have some practical limitations; for example, it is difficult to solve process synthesis problems involving complex chemical processes. Besides, sequential modular simulators like ASPEN, PROCESS, PROII, etc. are more widely used in chemical industries than equation oriented simulators. These simulators have grown in sophistication over the years and have useful capabilities to model many complex chemical processes. Therefore it is more desirable to have the process synthesis capability around these simulators. A new MINLP process synthesis capability built around the public version of ASPEN (Diwekar *et al.*, 1992) represents a step in this regard.

However, the MINLP approach to process synthesis also encounters some major difficulties which are outlined below:

1. The problem of implicit constraints is encountered in the case of MINLP synthesis in sequential modular simulators. This is because of the black box nature of the models in sequential modular simulators. Many of the MINLP environments are based on a two-level optimization algorithm consisting of an upper level MILP master problem and a lower level NLP problem. The MILP master problem predicts new discrete variables, while the NLP problem provides new continuous variables. The MILP master problem represents the linearized NLP problem with non-fixed binary variables since at each stage the MILP problem contains the linearization information from the NLP optimizer. When using a sequential modulator simulator, the implicit (black box) nature of the simulator results in the NLP optimizer being unable to transfer crucial linearization information to the master problem. This occurs because there is no derivative information from the Jacobian. As a consequence, the algorithm fails to find the optimal solution. Alternatives to make the implicit constraints explicit including methods of partitioning pseudo-decision variables are being developed (Diwekar and Rubin, 1992).

2. The MINLP approach requires the objective function to satisfy convexity conditions to guarantee convergence to the global optimum.

3. In synthesis, even fairly simple systems can lead to large combinatorial explosion. MINLP algorithms pose difficulties for large scale combinatorial problems where there are a large number of discrete decisions to be made. Standard problems without special structure that can be handled by MINLP solvers today typically have 40–60 binary variables (Grossmann, 1990).

4. The MINLP algorithms cannot handle large discontinuities in the solution space or infeasibility in the NLP stage. Although Duran and Grossmann (1986) presented a smooth approximation method to handle nondifferentiable MINLP synthesis problems, their approach is restricted to functional discontinuities at few specific points [e.g. the kink function, which is discontinuous at $f(x) = 0$]. For the synthesis problems where there are large-scale discontinuities such as the problem at hand, it is not possible to apply MINLP algorithms.

The alternative to MINLP process synthesis is to use simulated annealing. The advantages of simulated annealing in the chemical synthesis problem are that it is not a derivative based method and can handle large discontinuities in the state space much more easily. Simulated annealing is designed for large combinatorial problems. It has been successfully applied to the design of heat exchanger and pipeline networks (Dolan *et al.*, 1989). Also, it has been used in the scheduling of batch processes (Das *et al.*, 1990). However, to our knowledge, it has not been applied to the synthesis problem of a power plant, nor has it been incorporated into a sequential modular simulator program. We view the extension of simulated annealing to process synthesis in the same way that MINLP spread from network design problems to synthesis problems.

This paper extends the application of simulated annealing technique in two ways:

1. It incorporates simulated annealing into a chemical process simulator, SNAPS, thus allowing for the optimization of chemical flowsheet structures.
2. It is applied to the analysis of the design of a Brayton cycle space nuclear power plant.

In addition, a new method of representation of the objective function over the configuration space is presented.

PROCESS SYNTHESIS BY SIMULATED ANNEALING

Simulated annealing is a recently developed probabilistic method for combinatorial optimization based on ideas from statistical mechanics (Kirkpatrick *et al.*, 1983). It is a heuristic approach which is not guaranteed to find a global optimum unless the problem is convex. In these cases, simulated annealing is able to provide an approximate solution to NP-complete problems which is asymptotically exact (Das *et al.*, 1990). In practice, annealing yields a polynomial time solution to an exponential time problem. It has, however, proven to be a very powerful tool for many of the large scale technological problems facing engineers and scientists today. Though the seminal paper was published just 10 yr ago, a host of application areas have since developed (Johnson *et al.*, 1989). Simulated annealing has been applied to large scale combinatorial optimization problems such as VLSI design, pattern recognition-, and graph tearing with excellent results.

The analogy in simulated annealing 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 substance, 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 (Pr) of being in a state with energy E given by the Boltzmann distribution:

$$\Pr\{E=E\}=[1/Z(t)]\exp(-E/K_b T),$$

where K_b is Boltzmann's constant (1.3806×10^{-23} J/K) and $1/Z(t)$ is a normalization factor (Collins *et al.*, 1988).

In simulated annealing, the objective function (usually cost) becomes the energy of the system. The goal is to minimize the cost/energy. Simulating the behavior of the system then becomes a question of generating a random perturbation that displaces a "particle" (moving the system to another configuration). If the configuration that results from the move 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 criteria [accepted with probability $=\exp(-\Delta E/K_b T)$] (vanLaarhoven and Arts, 1987). This implies that at high temperatures, a large percentage of uphill moves are accepted. However, as the temperature gets colder, a small percentage of uphill moves are accepted. After the system has evolved to thermal equilibrium at a given temperature, the temperature is lowered and the annealing process continues until the system reaches a temperature that represents "freezing". Thus simulated annealing combines both iterative improvement in local areas and random jumping to help ensure the system does not get stuck in a local optimum. The general simulated annealing algorithm is given below (vanLaarhoven and Arts, 1987).

```
Initialize variables:
    $T_{initial}$, $T_{freeze}$, accept and reject limits or $N$
    (number of moves at a given temperature),
    initial configuration $S$ and cost ($S$).
While $T > T_{freeze}$ do
  While $i < N$ do
    Generate move $S'$ by perturbing $S$
    $\Delta cost = cost(S') - cost(S)$
    If $\Delta cost \leq 0$ or [random(0, 1)
       $< \exp(-\Delta cost/T)$] then accept $S'$,
       $S = S'$
    Update number of accepts, rejects
    Until equlibrium is reached at $T$ (when $i = N$)
    Update $T$: $T = \alpha T$
  Until $T < T_{freeze}$.
```

APPLICATION: SELECTING OPTIMAL DESIGN CONFIGURATIONS FOR A SPACE NUCLEAR POWER PLANT

The purpose of nuclear power in space is to fulfill the requirements for longevity of flight while minimizing the need for refueling. Space nuclear power technology involves the use of the thermal energy liberated by nuclear processes. The key factors which are important to space application of nuclear power plants include power and efficiency, mass and volume, reliability, safety and flexibility. A generic space nuclear power system consists of the energy source (isotope system or nuclear reactor), the primary heat transport system, an energy conversion technique, and a radiator for heat rejection. The current space nuclear power research program, SP-100, involves an assessment of a large number of combinations of reactors and power converters that could form a space nuclear power plant system. See Fig. 1 for possible types of systems.

Stringent spacecraft mass and volume restrictions force nuclear power systems designers to develop systems with the best power-to-mass ratio. Since nuclear reactors can be considered as a tremendous source of energy, the components which offer the best opportunity for mass and volume reductions are the thermal transport from the reactor to the power conversion system (PCS), the PCS and the radiator, not the nuclear reactor. There is no clear-cut choice between static and dynamic conversion processes; each application brings different considerations of factors such as power, mass, volume and reliability. Although earlier space nuclear power plant studies concentrated on static conversion techniques, dynamic cycles such as the Brayton cycle are being considered as plausible options for power conversion. This paper analyzes different configurations of Brayton cycle plants. Note that this is a subset of all the configurations given in Fig. 1.

As an example, we used SNAPS to analyze the Brayton cycle plant from the SP-100 evaluation report (Anderson *et al.*, 1983). This power plant is represented by chemical processes in Fig. 2. The

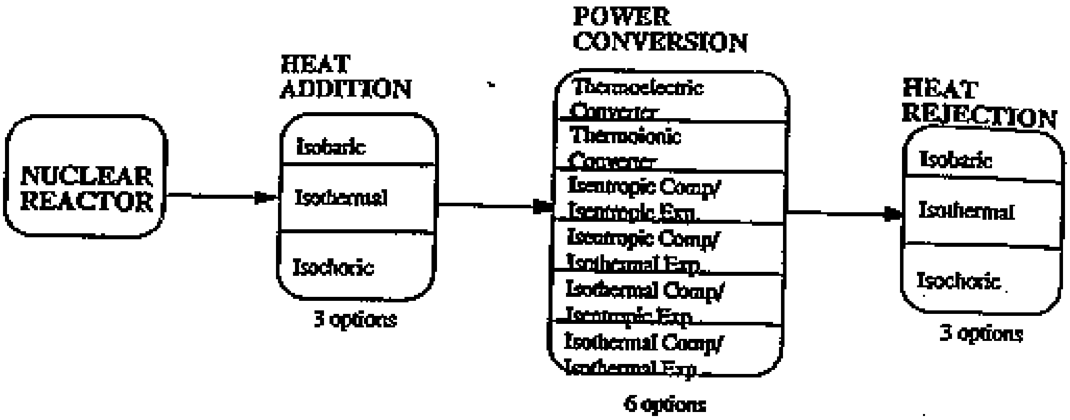

Fig. 1. Basic system configurations for space nuclear power plants.

power conversion stage in the Brayton cycle involves isentropic compression/expansion. He and Xe are the gases used in the system. See Fig. 3 for the SNAPS representation of this Brayton cycle. The overall flowsheet uses only three basic unit modules from SNAPS: compressors (CMP), heat exchangers (HEX) and the convergence block (ZTA). The converged results from SNAPS are shown in italics in Fig. 3.

IMPROVING PERFORMANCE BY COMBINATORIAL OPTIMIZATION

The performance of the Brayton cycle power conversion subsystem depends on the number of stages, and the design parameters such as the capacities of compressors and turbines (function of pressure ratio) and the capacities of heat exchangers (function of temperature difference or heat duty). A multistage expansion/compression system with intermediate heating and cooling, respectively, can increase the reversibility of the power conversion process resulting in higher power (see Fig. 4 for the power gained in a multistage compression/expansion cycle). Although this increases the efficiency of the Brayton cycle, it also increases the mass and cost of the system and decreases the reliability. The basic tradeoff is that adding compressors and turbines yields more power but increases the mass and cost.

Here we are considering the design alternatives to be number of stages, pressure ratios of turbines and compressors, and intermediate heat exchanger capacities. All the alternative configurations are embedded in a superstructure shown in Fig. 5. SNAPS

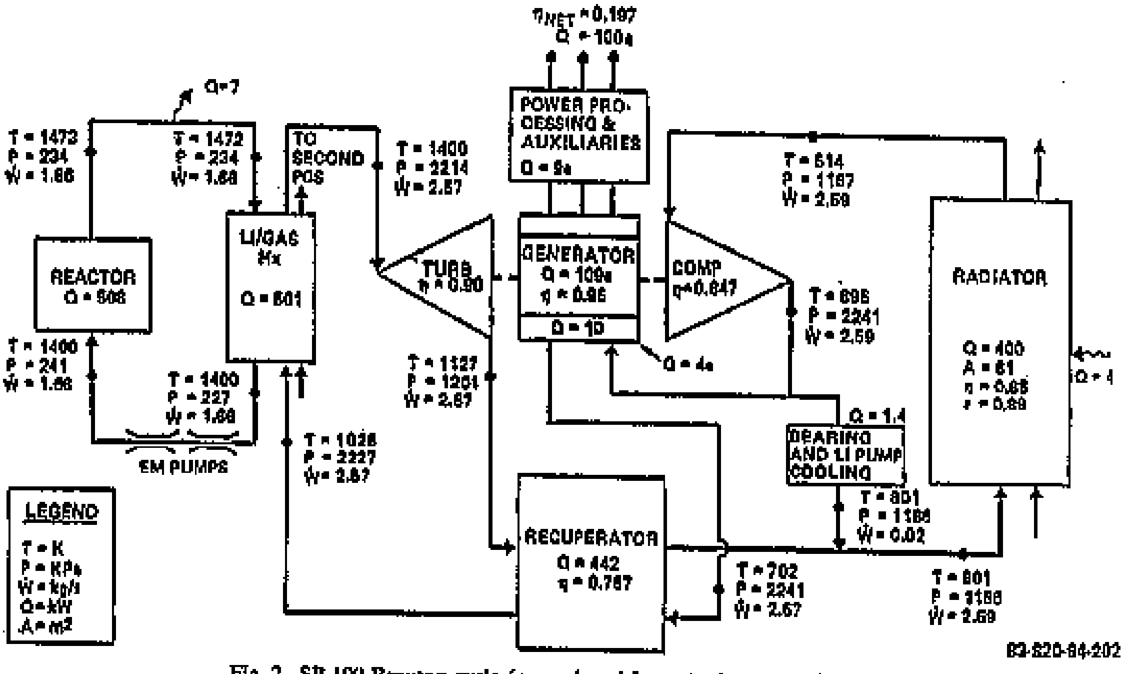

Fig. 2. SP-100 Brayton cycle (reproduced from Anderson *et al.*, 1983).

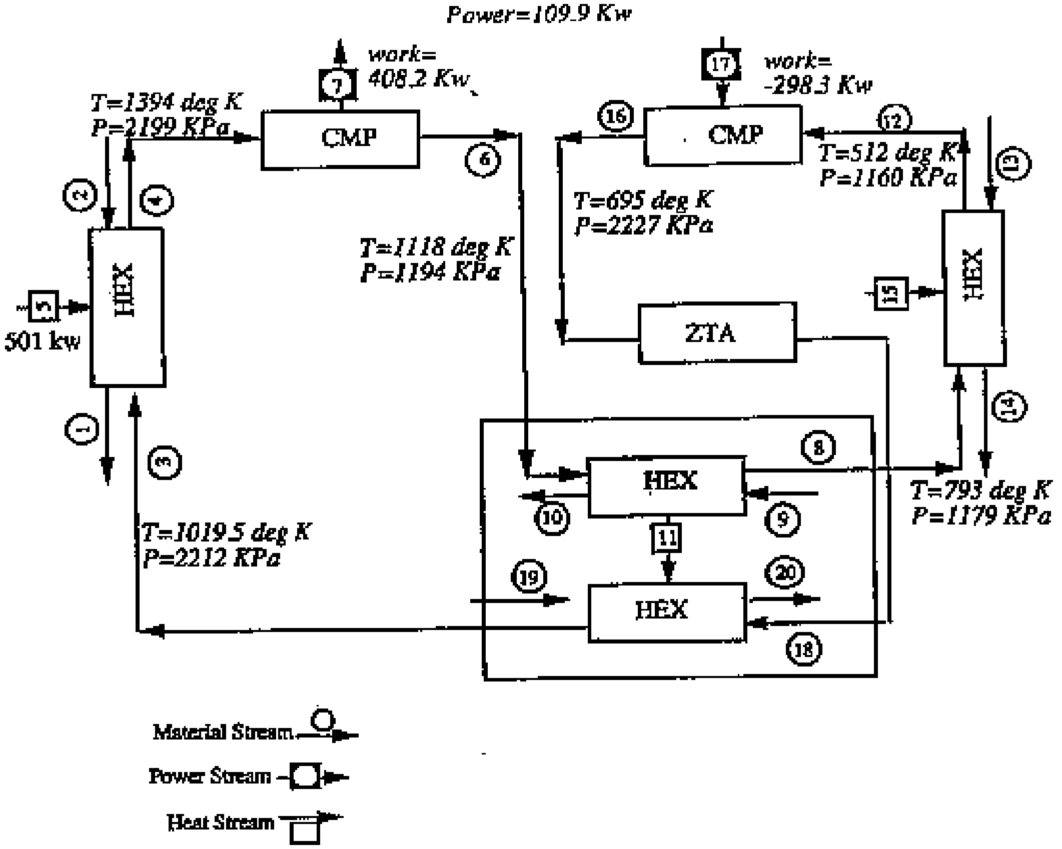

Fig. 3. SNAPS representation of the Brayton cycle with converged results.

calculates the energy balance and performance characteristics for one specific configuration given an input configuration and design parameters. However, the modularity of SNAPS allowed us to generate many such combinations easily. Figure 5 shows a system with five heat exchangers (HEX1–5), five turbines (TUR1–5), and five compressors (CMP1–5). This superstructure in Fig. 5 involves 26 unit modules and 67 streams. The radiator is shown in heat exchanger 8 and the heat addition from the nuclear reactor comes in heat exchanger 7. A set is defined as a group of unit modules consisting of one turbine, one compressor and one heat exchanger and is equivalent to one stage in compression with cooling and one stage in expansion with heating. For example, a set is turbine 3, compressor 3 and heat exchangers 3–3A in Fig. 5. The SNAPS model thus allowed us to consider Brayton cycles with 1–5 compressor/turbine/heat exchanger sets, with each unit having many options for parameter levels. The parameters used as decision variables for this problem were heat exchanger capacity (given by ΔT) and compressor/turbine pressure ratios. If a set is not selected in the annealing process, then for that heat exchanger the parameter ΔT is is equated to zero and the pressure ratio across the compressor and turbine equals 1. This allows the structure of the topology matrix to essentially remain constant no matter how many sets are considered.

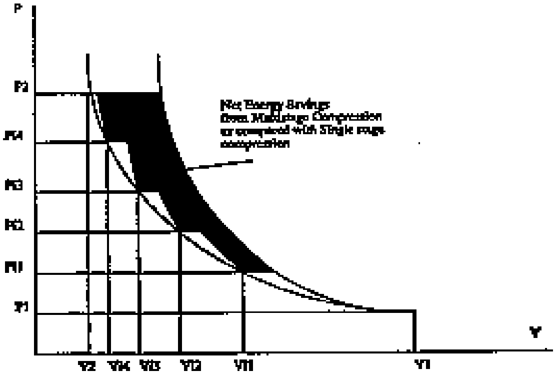

Fig. 4. Multistage compression cycle with interstage cooling.

Finding the overall optimal configuration (optimal both in terms of topological structure, that is, number and arrangement of components, and in terms of associated parameter levels per component) is a large-scale combinatorial optimization problem. First, there are many combinations of units such as heat exchangers, turbines and compressors. In additon, each component has a set of

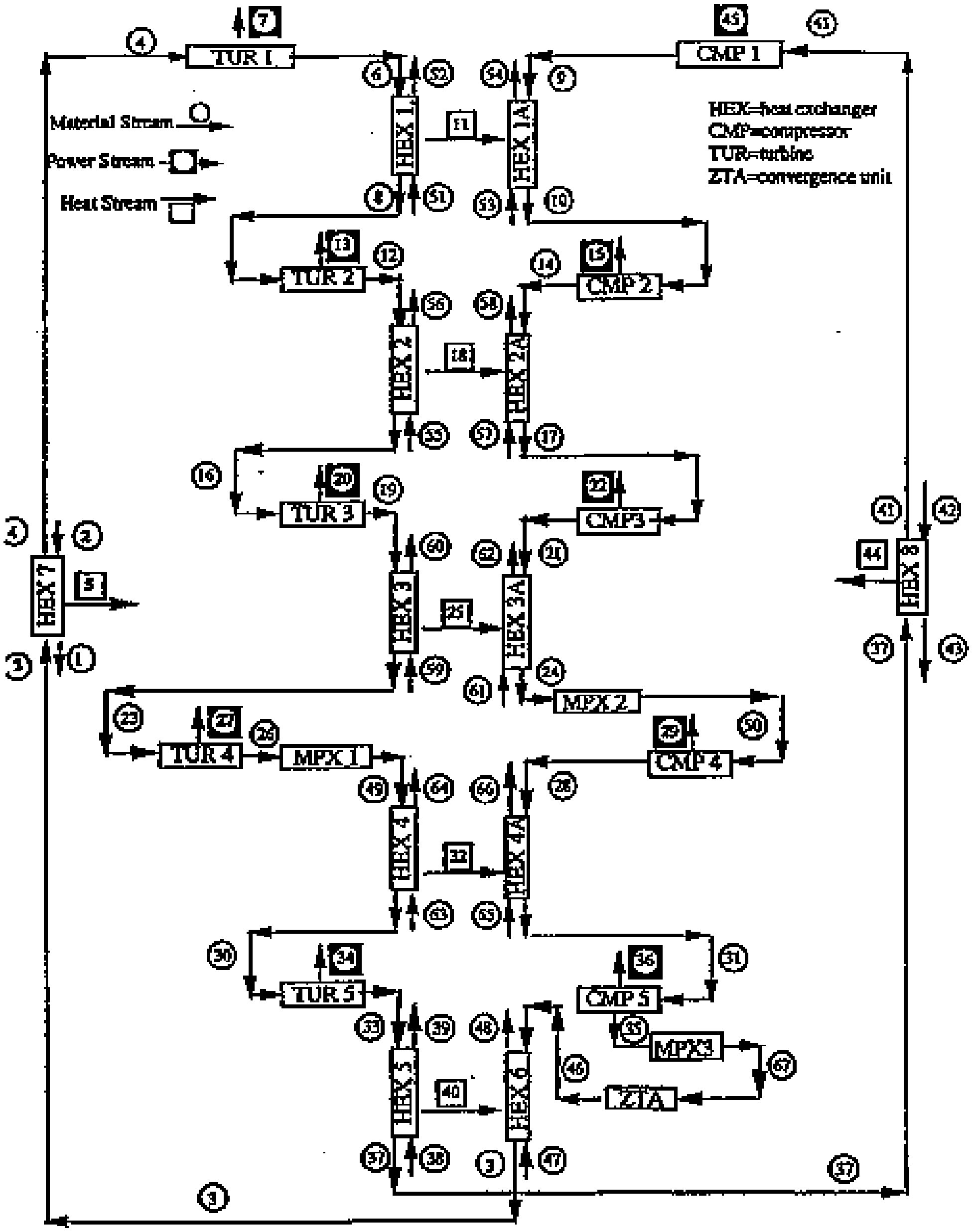

Fig. 5. A multistage expansion/compression Brayton cycle.

parameters which may vary across many discretized levels (for example, heat duty across a heat exchanger or pressure ratio of a turbine or compressor). For a Brayton cycle power plant consisting of n_{set} sets with n_{pr} pressure ratios and $n_{\Delta T}$ temperature levels, the number of combinations can be calculated as:

$$\text{total combinations} = \sum_{i=1}^{n_{set}} n^i_{pr} n^i_{\Delta T}.$$

Using the above formula, one can easily see that the number of combinations is very large; even in our simple model of a Brayton cycle system with up to five turbines, up to five compresssors, and up to five heat exchangers gives a state space of 10.2 *million* combinations. The method of simulated annealing was used to find the optimal structural topology and design parameter levels of the power plant.

IMPLEMENTATION

A major difficulty in the application of simulated annealing is defining the analogs to the entities in physical annealing. Specifically, it is necessary to specify the following: the configuration space, the cost function, the move generator (a method for randomly jumping for one configuration to another), the initial and final temperatures, the temperature decrement, and the equilibrium detection method. All of the above are dependent on problem structure. The initial and final temperatures in combination with the temperature decrement scheme and equilibrium detection method are generally referred to as the cooling schedule. Much work has been done in these areas for specific problems such as graph partitioning and VLSI design (Rutenbar 1989; Johnson *et al.*, 1989), but each problem is unique. What follows is a brief summary of the configuration space, cooling schedule, and move generator developed for space nuclear power plant optimization.

Configuration space

As discussed above, the configuration space in our model is a SNAPS representation consisting of unit module components such as turbines and heat exchangers. The model (Fig. 5) allows one to vary the number of sets and also the unit parameters across discretized levels. The parameters chosen were heat exchanger capacity and compressor/turbine pressure ratios. These were allowed to be set to one of five lev ls. Table 1 shows the different pressure ratios and heat exchanger capacities used in this problem.

Objective function

The objective function was defined as an additive function of the negative of power (because the objective is to maximize the power) and cost of heat exchangers, compressors and turbines. Weights were assigned to each of the components of the total cost function. The functional forms of the cost equations are:

$$C_{tur} = 2.984 P^{0.68},$$

$$C_{tot} = \omega_1 C_{tur} + \omega_2 P,$$

where P is power in kW, C_{tur} is the cost of a turbine in \$1000, C_{tot} is the total cost, and the weights are ω_1 and ω_2 (note the C_{tot} is a dimensionless quantity: it represents a multi-attribute weighted utility). The cost models for compressors/turbines were taken from DOE guidelines (*ASPEN Technical Reference Manual*, 1982) and can be replaced by more rigorous cost models if necessary. Although the objective function used for this study includes power and cost only, future studies will also include additional attributes such as mass, reliability and flexibility measures.

Table 1. Pressure ratios and temperature levels for the synthesis problem

Level	Heat duty ΔT (K)	Pressure ratio turbine/compressor
1	150 K	0.500/2.00
2	175 K	0.575/1.74
3	200 K	0.650/1.54
4	225 K	0.725/1.38
5	250 K	0.800/1.25

Initial temperature

If the initial temperature is too low, the search space is limited and the search becomes trapped in a local region. If thè initial temperature is too high, the algorithm spends a lot of time "boiling around" and wasting CPU time. To determine initial temperature, the following criteria were satisfied [the idea is to initially have a high percentage of moves which are accepted (Kirkpatrick *et al.*, 1983)]:

1. Take an initial temperature $T_{init} > 0$.
2. Perform $N_{samp} (= 100)$ sample moves according to the annealing schedule.
3. If the acceptable moves are $< 80\%$ of the total sampled ($M_{acc}/N_{samp} < 0.8$) then set $T_{init} = 2^* T_{init}$ and go back to (2).
4. If the acceptable moves are $> 80\%$ of the total sampled ($M_{acc}/N_{samp} > 0.8$) then the initial temperature for the simulated annealing is T_{init}.

Final temperature

The final temperature was chosen so that the algorithm stopped after 10 successive temperature decrements with no change in the optimal configuration.

Equilibrium detection and temperature decrement

If the temperature decrement is too big, the algorithm quickly "quenches" and could get stuck in a local minimum with not enough thermal energy to climb out. On the other hand, if the temperature decrement is very small, much CPU time is required. Some rules for setting the new temperature at each level are:

1. $T_{new} = \alpha T_{old}$, where $0.8 <= \alpha <= 0.99$.
2. $T_{new} = T_{old}[1 + (1+\gamma) T_{old}/3\sigma]^{-1}$.
   This annealing schedule was developed by Aarts and van Laarhoven and is based on the idea of maintaining quasi-equilibrium at each temperature (van Laarhoven and Aarts, 1987). σ is the standard deviation of the cost at the annealing temperature T_{old} and γ is a parameter which governs the speed of annealing (usually very small).
3. $T_{new} = T_{old} \exp(\text{ave}\ \Delta\ \text{cost} \times T_{old}/\sigma^2)$.
   This schedule was developed by Huang and is based upon the idea of controlling the average change in cost at each step instead of taking a fixed change in the log T (as in Schedule 1).

This allows one to take more moves in regions of lower variability, so that one takes many small steps at cooler temperatures when σ is low (Huang *et al.*, 1986).

Schedule 1 was chosen for decrementing temperature with $\alpha=0.9$.

Simulated annealing needs to reach quasi-equilibrium at each state or it is not truly annealing. It is difficult to detect equilibrium but there are some crude methods:

1. Set N= number of states visited at each temperature.
2. Set an accept/reject ratio of 1:10.
3. Sample N moves. Assuming the change in cost has a Gaussian distribution, the number of samples within the region (average cost $+\sigma/2$) is erlang(0.5)=0.38. Thus if 38% of the samples fall within the accept region before 62% of the samples fall outside, one assumes the system has reached equilibrium (Huang *et al.*, 1986).

In practice, it rarely is possible or efficient to run to an equilibirium distribution at intermediate temperatures. The algorithm used visited a fixed number of states at each temperature (that is, the lower bound on the total number of moves N was given by the number of successful moves, N_{succ}), but also allowed for a number of accepts and rejects in the 1:10 ratio (the algorithm drops to the next temperature when the number of accepts or rejects reach their limits).

Move generator

The creation of a move generator is difficult because a move needs to be "random" yet result in a configuration which is in the vicinity of the previous configuration. Based upon an annealing algorithm for a heat exchanger network (Dolan *et al.*, 1989) and exhaustive experimentation, an optimal move generator was created such that each move could result in one of the following permutations of the current configuration:

1. Add a *random number* of sets. A set is one heat exchanger, one compressor and one turbine. Randomly set the parameters of the added sets to one of the possible levels.
2. Delete a *random number* of sets.
3. Remain at the same number of sets, but "bump" one of the parameters up or down by a *random number* of levels (not exceeding the maximum allowed level). When the temperature gets small enough, however, limit the move size to plus or minus one level.

The above move possibilities were weighted 10:10:80 (that is, 80% of the time the move involves only changing a parameter level, not the overall topology structure). This move generator was one of three used in this study and gives the best results because it is the most "randomized" one. It was found that bumping the parameters or number of sets up or down by one level was not sufficient for a comprehensive search of the configuration space. The large discontinuities in the solution space caused the failure of the annealing algorithm when small jumps such as bumping parameter or number sets up or down by one level were used. In this case even choosing different annealing parameter (e.g. α closer to unity) did not help to arrive at the optimum objective function.

RESULTS

Solution space as a distribution function (DF)

The annealing algorithm described above is coded in FORTRAN around the SNAPS power plant simulator. The overview of the program structure is shown in Fig. 6.

To prove that the annealing algorithm does converge upon a (near) optimal solution, it was necessary to generate the entire space of possible configurations and their associated weighted cost functions. Because this solution space is a nonconvex, discrete combinatorial space it was not possible to analytically take derivatives to find the optimum. The only method available was "brute force", generating each combination. This took a very large amount of computation time: to generate the 10,172,525 combinations took 233,338.29 CPU s or 2.7 days on the Cray Y-MP.†

Since the number of decision variables is more than three, it is difficult to visualize the solution surface as a function of the possible design options and parameter settings. The distribution functions can be used to represent the complex surface and to provide some insight into the nature of the solution surface. For example, the stiffness of the cumulative distribution function (CDF) or the hills of the distribution function (DF) represent the flatness of the solution space, indicating a higher probability of getting trapped in local optima in these regions. This is a novel way to represent the solution surface.

The cumulative distribution of the solution space generated by exhaustive search is shown in Fig. 7.‡ This CDF shows that most of the combinations are

† Note: the code was not vectorized, so it was not able to run very efficiently.

‡ Note that the cost function shown in Figs 7 and 8 is given in absolute value for ease of representation.

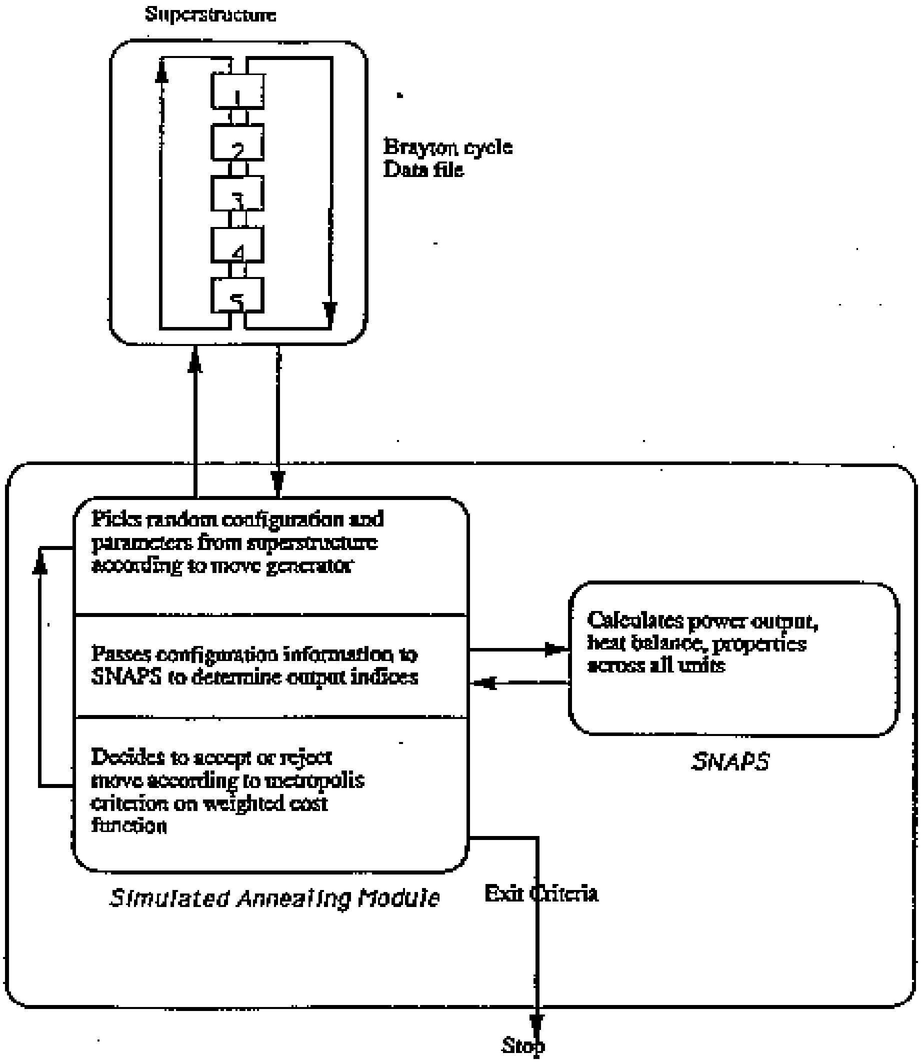
Superstructure
1
2
3
4
5
Brayton cycle
Data file
Picks random configuration and parameters from superstructure according to move generator
Passes configuration information to SNAPS to determine output indices
Decides to accept or reject move according to metropolis criterion on weighted cost function
Calculates power output, heat balance, properties across all units
SNAPS
Simulated Annealing Module
Exit Criteria
Stop

Fig. 6. Overview of program structure.

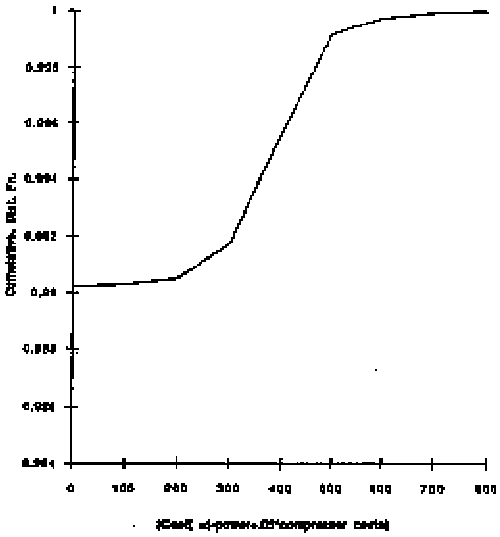

Fig. 7. Cumulative distribution function of the entire solution space.

infeasible. Over 99% of the combinations are infeasible which introduces large discontinuities, but it is not possible to know which ones prior to running the combinations through SNAPS. The large number of infeasibilities is due to the violation of physical constraints in the Brayton cycle (for example, the radiator temperature being out of bounds). Note that simulated annealing does not have a way of dealing explicitly with constraint enforcement. We chose to implement constraint enforcement through the use of a penalty function. The power output and cost function were set to zero if the specific configuration was infeasible. It was not possible to use a more complex penalty function because in the case of Brayton cycle power plant it was difficult to close the energy balance when the constraints were violated. Though our penalty function was very crude due to the nature of infeasibilities, we found simulated annealing robust and able to incorporate the

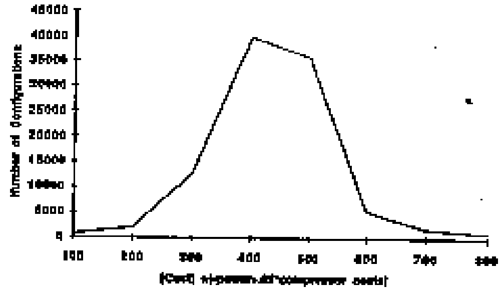

Fig. 8. Distribution function of the entire solution space.

penalty function costs into the algorithm. This allowed us to handle large numbers of constraints.

It can be seen that simulated annealing performs very well even in the face of a large number of discontinuities. Of the approximately 100,000 combinations that are feasible, the majority fall in the cost range of −400 to −500 (see the distribution function in Fig. 8). The potential for getting stuck in a local optima in this range is high because most of the feasible solutions are in this region. There are 2835 combinations with a cost below −600. This represents 0.02% of the possible solutions.

Optimal design configurations

The annealing code was run with many different values of the annealing parameters such as initial temperature ($T_{\rm init}$), temperature decrement (α), number of successful moves at each temperature ($N_{\rm succ}$) and the random number generator (iseed) for the case of the 1000 kW Brayton cycle plant with up to five compression/expansion stages. The results are shown in Table 2. Note that all the optimal configurations have three or four sets. The optimal levels for heat duty and pressure ratio vary, but many are at the high end (250° for heat duty, 0.8 for pressure ratio). The power output ranges from 931 to 993 kW.† The optimum weighted cost function ranges from −660 to −729. Note that the reason the annealing algorithm does not converge to the same solution each time is because the optimal solutions are spread out in a "gopher hole" fashion in the solution space. Since the surface has number of local minima it is interesting to see effect of different annealing parameters on the performance in detail. Figures 9 and 10 show effect of temperature decrement α which governs the cooling schedule and number of successful moves $N_{\rm succ}$ which decides the equilibrium at each temperature, on the optimal solution and on the CPU time requirements. It can be seen that if slower schedule (bigger α) is used then CPU time is higher as expected, however the cooling schedule does not affect the solution that much. The effect on the optimal solution is more pronounced for equilibrium detection criteria, the larger $N_{\rm succ}$ always gives better solution at the cost of increased CPU time. Both these effects can be attributed easily to the solution surface we are dealing with. The solution surface is flat at many places because of the large discontinuities and infeasibilities where finding an uphill or downhill move is not easy unless one takes large number of random moves at each temperature. Therefore, the equilibrium detection requires more samples and the solution improves for larger value of $N_{\rm succ}$. Similarly, one encounters very few uphill moves on this surface and hence the effect of cooling schedule which decides acceptance or rejection of uphill moves is not that significant.

The overall performance of the annealing program is excellent; all solutions ended up above the 99.98% point on the CDF. Thus simulated annealing avoids getting stuck in a local optima in the −400 to −500 region. Instead, it converges in the "optimal" region, the upper 0.02% of the CDF. Simulated annealing converges on a solution after only sampling a small amount of configurations, on average 11,3375 (this is 0.1% of the possible solutons). Note that the "worst" optimum found, −660, is 9.5% away from the best optimum found, −729. A single run of the annealing program takes approx 1/1000 of the time it took to generate the entire configuration space. Thus annealing provides significant computational advantage over exhaustive search.

CONCLUSION

Simulated annealing is a powerful technique for solving large combinatorial optimization problems. It has been successfully applied to the case of selecting optimal design configurations for a Brayton cycle space nuclear power plant. This is the first time that such a technique has been applied to nuclear power plants, and also the first time it has been implemented in a sequential modular simulator for chemical processes. The Brayton cycle power plant problem presented in this paper is associated with large discontinuities in the functional space and is suitable for application of simulated annealing. Simulated annealing is a trade-off among the speed of convergence, the probability of being trapped in local optima, the handling of discontinuities, and the

† Note that the feasibility limitations of temperature in this case are $T_{\rm sink} = 0$ K, which is not possible in practice. This problem represents an idealized solution that is theoretically achievable with multi-stage compression/expansion. The efficiency of the idealized Brayton cycle is given by the Carnot limit for dynamic conversion cycles: $\eta_{\rm Carnot} = (T_{\rm source} - T_{\rm sink})/T_{\rm source}$.

Table 2. Optimal design configurations for different annealing schedules

Configuration 1

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.8
$N_{\rm succ}=100$	250	0.725
$T_{\rm init}=100$	250	0.5
iseed = 1		
Number of sets:	3	
Power output:	−958 kW	
Optimum cost:	−724	

Configuration 2

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.8
$N_{\rm succ}=100$	225	0.5
$T_{\rm init}=100$	250	0.575
iseed = 5		
Number of sets:	3	
Power output:	−931 kW	
Optimum cost:	−660	

Configuration 3

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.575
$N_{\rm succ}=100$	250	0.725
$T_{\rm init}=100$	175	0.725
iseed = 234	250	0.8
Number of sets:	4	
Power output:	−984 kW	
Optimum cost:	−700	

Configuration 4

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.725
$N_{\rm succ}=100$	225	0.65
$T_{\rm init}=100$	175	0.8
iseed = 7787	225	0.725
Number of sets:	4	
Power output:	−978 kW	
Optimum cost:	−705	

Configuration 5

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.725
$N_{\rm succ}=100$	175	0.65
$T_{\rm init}=200$	250	0.8
iseed = 1	225	0.725
Number of sets:	4	
Power output:	−981 kW	
Optimum cost:	−705	

Configuration 6

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.9$	—	0.575
$N_{\rm succ}=100$	175	0.725
$T_{\rm init}=300$	250	0.8
iseed = 1	250	0.725
Number of sets:	4	
Power output:	−980 kW	
Optimum cost:	−688	

Configuration 7

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.95$	—	0.725
$N_{\rm succ}=100$	250	0.65
$T_{\rm init}=200$	225	0.8
iseed = 7787	225	0.8
Number of sets:	4	
Power output:	−993 kW	
Optimum cost:	−729	

Configuration 8

	Heat duty ΔT(K)	Pressure ratio
$\alpha=0.95$	—	0.8
$N_{\rm succ}=200$	225	0.65
$T_{\rm init}=200$	225	0.8
iseed = 1	175	0.725
Number of sets:	4	
Power output:	−986 kW	
Optimum cost:	−723	

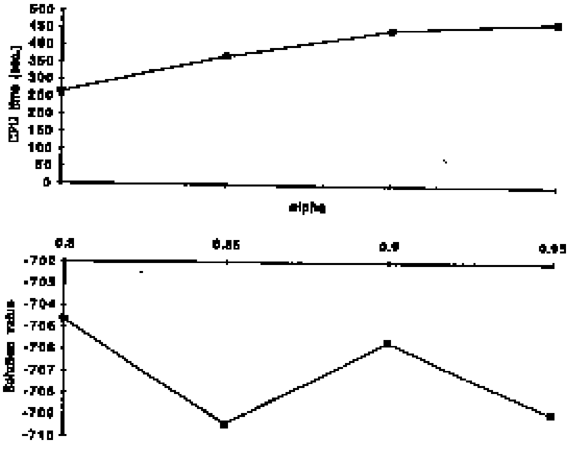

Fig. 9. Effect of changing cooling schedule on annealing performance.

ability to deal with constraints. Although gradient techniques are faster in cases of convex, continuous objective functions over convex search spaces, annealing provides many advantages in problems such as process synthesis problems where the search space is nonconvex and discontinuous and the objective function is nonconvex. In these cases, annealing avoids the problem of local optima: it provides one with a set of optima. In addition, it is able to handle large discontinuities and infeasibilities due to constraint violations that gradient-based techniques may simply not be able to do.

Performance of the annealing algorithm is highly dependent on the annealing schedule and its implementation. Two major questions are: how to determine what sorts of combinatorial optimization prob-

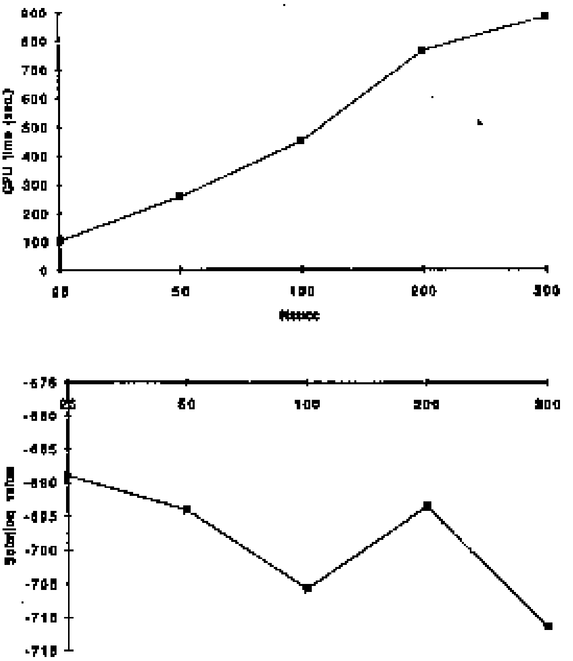

Fig. 10. Effect of changing equilibrium detection criteria on annealing performance.

lems lend themselves to annealing, and how to guarantee that the move generator and cooling schedule are properly sampling the space for annealing to occur. "Not all combinatoric optimization problems can be annealed to give satisfactory solutions . . . a mostly flat landscape with numerous, densely packed gopher holes . . . is probably impossible to anneal" (Rutenbar, 1989). This search has found that a basic move generator which randomly bumps a parameter up or down one level is not sufficient for simulated annealing of the type of cost functions (often characterized by large flat spaces) appropriate to Brayton cycle space nuclear power plant designs. Thus a more robust move generator which randomly changes the number of paired sets and moves the parameters a random number of levels was developed. The space power plant configuration problem has provided an innovative area in which to apply simulating annealing. In addition, a novel interpretation of the distribution functions have been used to represent the solution surface. The future work will address the problem of optimizing the simulated annealing algorithm itself.

Acknowledgements—The authors would like to thank Dr Steve Zitney and Cray Research, Inc. for allowing the use of their Y-MP machine. This work has been partially funded by a DOE Computational Science Graduate Student Fellowship.

REFERENCES

Anderson R. V. *et al.*, SP-100 program space reactor system and subsystem investigation. ESG-DOE-13413 (DEE855009136), Rockwell International, Canoga Park, CA (1983).

ASPEN Technical Reference Manual, DOE Publication 16481-1202, Vol. 2, Morgantown Energy Technology Center, Morgantown, VA (1982).

Collis N. E., R. W. Eglese and B. L. Golden, Simulated annealing—an annotated bibliography. *Am. J. Math. Mgmt Sci.* **8**, 209–307 (1988).

Das H., P. T. Cummings and M. D. LeVan, Scheduling of serial multiproduct batch processes via simulated annealing. *Computers chem. Engng* **14**, 1351–1362 (1990).

Diwekar U. M., I. E. Grossmann and E. S. Rubin, An MINLP process synthesizer for a sequential modular simulator. *Ind. Engng Chem. Res.* **31**, American Chemical Society, pp. 313–323 (1992).

Diwekar U. M. and E. S. Rubin, An efficient handling of the implicit constraints problem for the Aspen MINLP synthesizer, *Ind. Engng Chem. Res.* **32**, 2006–2011 (1993).

Diwekar U. M. and B. Morel, Space nuclear power plants—I. *Computers chem. Engng* **17**, 873–878 (1993).

Dolan W. B., P. T. Cummings and M. D. LeVan, Process optimization via simulated annealing: application to network design. *AIChE Jl* **35**, 725–736 (1985).

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

Duran M. A. and I. E. Grossmann, Simultaneous optimization and heat integration of chemical processes. *AIChE Jl* **32**, 123–138 (1986).

Grossmann I. E., Mixed-integer nonlinear programming techniques for the synthesis of engineering systems. *Res. Engng Des.* **1**, 205–228 (1990).

Gupta O. K., Branch and bound experiments in nonlinear integer programming. Ph.D. Thesis, Purdue University (1980).

Huang M. D., F. Romeo and A. L. Sangiovanni-Vincentelli, An efficient general cooling schedule for simulating annealing. *Proc. IEEE Int. Conf. on Computer Design*, 381–384 (1986).

Hughes R. R., R. K. Malik and L. T. Biegler, SPAD: simulator for process analysis and design. University of Wisconsin-Madison (1981).

Johnson D., C. Aragon, L. McGeoch and C. Schevon, Optimization by simulated annealing: an experimental evaluaton: Part I, graph partitioning. *Ops Res.* **37**, 865–892 (1989).

Kirkpatrick S., C. Gelatt and M. Vecchi, Optimization by simulated annealing. *Science* **220**, 670–680 (1983).

Kocis G. R. and I. E. Grossmann, Computational experience with DICOPT solving MINLP problems in process synthesis engineering. *Computers chem. Engng* **13**, 797–819.

Kravanja Z. and I. E. Grossmann, PROSYN—an MINLP process synthesizer. *Computers chem. Engng* **14**, 1363 (1990).

Paules G. E. and C. A. Floudas, 'APROS: algorithmic development methodology for discrete-continuous optimization problems. *Ops Res. J.* **37**, 902 (1989).

Press W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, pp. 328–333. Cambridge University Press, New York (1987).

Rutenbar R. A., Simulated annealing algorithms: an overview. *IEEE Circuits Dev. Mag.* **100**, 19–26 (1989).

vanLaarhoven P. J. M. and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Reidel Publishing Co., Holland (1987).

Viswanathan J. and I. E. Grossmann, A combined penalty function and outer approximation method for MINLP optimization, *Computers chem. Engng* **14**, 769 (1990).

Westerberg A. W., H. P. Hutchison, R. L. Motard and P. Winter, *Process Flowsheeting*. Cambridge University Press, Cambridge (1979).