13 May 1983, Volume 220, Number 4598 SCIENCE

Optimization by Simulated Annealing

S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi

Summary. There is a deep and useful connection between statistical mechanics (the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature) and multivariate or combinatorial optimization (finding the minimum of a given function depending on many parameters). A detailed analogy with annealing in solids provides a framework for optimization of the properties of very large and complex systems. This connection to statistical mechanics exposes new information and provides an unfamiliar perspective on traditional optimization problems and methods.

In this article we briefly review the central constructs in combinatorial optimization and in statistical mechanics and then develop the similarities between the two fields. We show how the Metropolis algorithm for approximate numerical simulation of the behavior of a many-body system at a finite temperature provides a natural tool for bringing the techniques of statistical mechanics to bear on optimization.

We have applied this point of view to a number of problems arising in optimal design of computers. Applications to partitioning, component placement, and wiring of electronic systems are described in this article. In each context, we introduce the problem and discuss the improvements available from optimization.

Of classic optimization problems, the traveling salesman problem has received the most intensive study. To test the power of simulated annealing, we used the algorithm on traveling salesman problems with as many as several thousand cities. This work is described in a final section, followed by our conclusions.

Combinatorial Optimization

The subject of combinatorial optimization (*1*) consists of a set of problems that are central to the disciplines of computer science and engineering. Research in this area aims at developing efficient techniques for finding minimum or maximum values of a function of very many independent variables (*2*). This function, usually called the cost function or objective function, represents a quantitative measure of the "goodness" of some complex system. The cost function depends on the detailed configuration of the many parts of that system. We are most familiar with optimization problems occurring in the physical design of computers, so examples used below are drawn from that context. The number of variables involved may range up into the tens of thousands.

The classic example, because it is so simply stated, of a combinatorial optimization problem is the traveling salesman problem. Given a list of N cities and a means of calculating the cost of traveling between any two cities, one must plan the salesman's route, which will pass through each city once and return finally to the starting point, minimizing the total cost. Problems with this flavor arise in all areas of scheduling and design. Two subsidiary problems are of general interest: predicting the expected cost of the salesman's optimal route, averaged over some class of typical arrangements of cities, and estimating or obtaining bounds for the computing effort necessary to determine that route.

All exact methods known for determining an optimal route require a computing effort that increases exponentially with N, so that in practice exact solutions can be attempted only on problems involving a few hundred cities or less. The traveling salesman belongs to the large class of NP-complete (nondeterministic polynomial time complete) problems, which has received extensive study in the past 10 years (*3*). No method for exact solution with a computing effort bounded by a power of N has been found for any of these problems, but if such a solution were found, it could be mapped into a procedure for solving all members of the class. It is not known what features of the individual problems in the NP-complete class are the cause of their difficulty.

Since the NP-complete class of problems contains many situations of practical interest, heuristic methods have been developed with computational requirements proportional to small powers of N. Heuristics are rather problem-specific: there is no guarantee that a heuristic procedure for finding near-optimal solutions for one NP-complete problem will be effective for another.

There are two basic strategies for heuristics: "divide-and-conquer" and iterative improvement. In the first, one divides the problem into subproblems of manageable size, then solves the subproblems. The solutions to the subproblems must then be patched back together. For this method to produce very good solutions, the subproblems must be naturally disjoint, and the division made must be an appropriate one, so that errors made in patching do not offset the gains

S. Kirkpatrick and C. D. Gelatt, Jr., are research staff members and M. P. Vecchi was a visiting scientist at IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598. M. P. Vecchi's present address is Instituto Venezolano de Investigaciones Cientificas, Caracas 1010A, Venezuela.

obtained in applying more powerful methods to the subproblems (*4*).

In iterative improvement (*5, 6*), one starts with the system in a known configuration. A standard rearrangement operation is applied to all parts of the system in turn, until a rearranged configuration that improves the cost function is discovered. The rearranged configuration then becomes the new configuration of the system, and the process is continued until no further improvements can be found. Iterative improvement consists of a search in this coordinate space for rearrangement steps which lead downhill. Since this search usually gets stuck in a local but not a global optimum, it is customary to carry out the process several times, starting from different randomly generated configurations, and save the best result.

There is a body of literature analyzing the results to be expected and the computing requirements of common heuristic methods when applied to the most popular problems (*1–3*). This analysis usually focuses on the worst-case situation—for instance, attempts to bound from above the ratio between the cost obtained by a heuristic method and the exact minimum cost for any member of a family of similarly structured problems. There are relatively few discussions of the average performance of heuristic algorithms, because the analysis is usually more difficult and the nature of the appropriate average to study is not always clear. We will argue that as the size of optimization problems increases, the worst-case analysis of a problem will become increasingly irrelevant, and the average performance of algorithms will dominate the analysis of practical applications. This large number limit is the domain of statistical mechanics.

Statistical Mechanics

Statistical mechanics is the central discipline of condensed matter physics, a body of methods for analyzing aggregate properties of the large numbers of atoms to be found in samples of liquid or solid matter (*7*). Because the number of atoms is of order 10^{23} per cubic centimeter, only the most probable behavior of the system in thermal equilibrium at a given temperature is observed in experiments. This can be characterized by the average and small fluctuations about the average behavior of the system, when the average is taken over the ensemble of identical systems introduced by Gibbs. In this ensemble, each configuration, defined by the set of atomic positions, $\{r_i\}$, of the system is weighted by its Boltzmann probability factor, $\exp(-E(\{r_i\})/k_BT)$, where $E(\{r_i\})$ is the energy of the configuration, k_B is Boltzmann's constant, and T is temperature.

A fundamental question in statistical mechanics concerns what happens to the system in the limit of low temperature—for example, whether the atoms remain fluid or solidify, and if they solidify, whether they form a crystalline solid or a glass. Ground states and configurations close to them in energy are extremely rare among all the configurations of a macroscopic body, yet they dominate its properties at low temperatures because as T is lowered the Boltzmann distribution collapses into the lowest energy state or states.

As a simplified example, consider the magnetic properties of a chain of atoms whose magnetic moments, μ_i, are allowed to point only "up" or "down," states denoted by $\mu_i = \pm 1$. The interaction energy between two such adjacent spins can be written $J\mu_i\mu_{i+1}$. Interaction between each adjacent pair of spins contributes $\pm J$ to the total energy of the chain. For an N-spin chain, if all configurations are equally likely the interaction energy has a binomial distribution, with the maximum and minimum energies given by $\pm NJ$ and the most probable state having zero energy. In this view, the ground state configurations have statistical weight $\exp(-N/2)$ smaller than the zero-energy configurations. A Boltzmann factor, $\exp(-E/k_BT)$, can offset this if k_BT is smaller than J. If we focus on the problem of finding empirically the system's ground state, this factor is seen to drastically increase the efficiency of such a search.

In practical contexts, low temperature is not a sufficient condition for finding ground states of matter. Experiments that determine the low-temperature state of a material—for example, by growing a single crystal from a melt—are done by careful annealing, first melting the substance, then lowering the temperature slowly, and spending a long time at temperatures in the vicinity of the freezing point. If this is not done, and the substance is allowed to get out of equilibrium, the resulting crystal will have many defects, or the substance may form a glass, with no crystalline order and only metastable, locally optimal structures.

Finding the low-temperature state of a system when a prescription for calculating its energy is given is an optimization problem not unlike those encountered in combinatorial optimization. However, the concept of the temperature of a physical system has no obvious equivalent in the systems being optimized. We will introduce an effective temperature for optimization, and show how one can carry out a simulated annealing process in order to obtain better heuristic solutions to combinatorial optimization problems.

Iterative improvement, commonly applied to such problems, is much like the microscopic rearrangement processes modeled by statistical mechanics, with the cost function playing the role of energy. However, accepting only rearrangements that lower the cost function of the system is like extremely rapid quenching from high temperatures to $T = 0$, so it should not be surprising that resulting solutions are usually metastable. The Metropolis procedure from statistical mechanics provides a generalization of iterative improvement in which controlled uphill steps can also be incorporated in the search for a better solution.

Metropolis *et al.* (*8*), in the earliest days of scientific computing, introduced a simple algorithm that can be used to provide an efficient simulation of a collection of atoms in equilibrium at a given temperature. In each step of this algorithm, an atom is given a small random displacement and the resulting change, ΔE, in the energy of the system is computed. If $\Delta E \leq 0$, the displacement is accepted, and the configuration with the displaced atom is used as the starting point of the next step. The case $\Delta E > 0$ is treated probabilistically: the probability that the configuration is accepted is $P(\Delta E) = \exp(-\Delta E/k_BT)$. Random numbers uniformly distributed in the interval (0,1) are a convenient means of implementing the random part of the algorithm. One such number is selected and compared with $P(\Delta E)$. If it is less than $P(\Delta E)$, the new configuration is retained; if not, the original configuration is used to start the next step. By repeating the basic step many times, one simulates the thermal motion of atoms in thermal contact with a heat bath at temperature T. This choice of $P(\Delta E)$ has the consequence that the system evolves into a Boltzmann distribution.

Using the cost function in place of the energy and defining configurations by a set of parameters $\{x_i\}$, it is straightforward with the Metropolis procedure to generate a population of configurations of a given optimization problem at some effective temperature. This temperature is simply a control parameter in the same units as the cost function. The simulated annealing process consists of first "melting" the system being optimized at a high effective temperature, then lower-

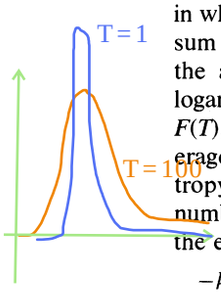
ing the temperature by slow stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach a steady state. The sequence of temperatures and the number of rearrangements of the $\{x_i\}$ attempted to reach equilibrium at each temperature can be considered an annealing schedule.

Annealing, as implemented by the Metropolis procedure, differs from iterative improvement in that the procedure need not get stuck since transitions out of a local optimum are always possible at nonzero temperature. A second and more important feature is that a sort of adaptive divide-and-conquer occurs. Gross features of the eventual state of the system appear at higher temperatures; fine details develop at lower temperatures. This will be discussed with specific examples.

Statistical mechanics contains many useful tricks for extracting properties of a macroscopic system from microscopic averages. Ensemble averages can be obtained from a single generating function, the partition function, Z,

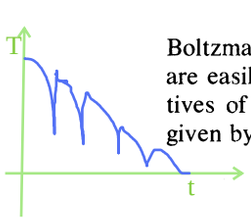
$$Z = \mathrm{Tr} \exp\left(\frac{-E(x)}{k_B T}\right) \quad (1)$$

E(x) ->



in which the trace symbol, Tr, denotes a sum over all possible configurations of the atoms in the sample system. The logarithm of Z, called the free energy, $F(T)$, contains information about the average energy, $\langle E(T) \rangle$, and also the entropy, $S(T)$, which is the logarithm of the number of configurations contributing to the ensemble at T:

$$-k_B T \ln Z = F(T) = \langle E(T) \rangle - TS \quad (2)$$



Boltzmann-weighted ensemble averages are easily expressed in terms of derivatives of F. Thus the average energy is given by

$$\langle E(T) \rangle = \frac{-d \ln Z}{d(1/k_B T)} \quad (3)$$

and the rate of change of the energy with respect to the control parameter, T, is related to the size of typical variations in the energy by

$$C(T) = \frac{d \langle E(T) \rangle}{dT} = \frac{[\langle E(T)^2 \rangle - \langle E(T) \rangle^2]}{k_B T^2} \quad (4)$$

In statistical mechanics $C(T)$ is called the specific heat. A large value of C signals a change in the state of order of a system, and can be used in the optimization context to indicate that freezing has begun and hence that very slow cooling is required. It can also be used to determine the entropy by the thermodynamic relation

$$\frac{dS(T)}{dT} = \frac{C(T)}{T} \quad (5)$$

Integrating Eq. 5 gives

$$S(T) = S(T_1) - \int_T^{T_1} \frac{C(T')\, dT}{T} \quad (6)$$

where T_1 is a temperature at which S is known, usually by an approximation valid at high temperatures.

The analogy between cooling a fluid and optimization may fail in one important respect. In ideal fluids all the atoms are alike and the ground state is a regular crystal. A typical optimization problem will contain many distinct, noninterchangeable elements, so a regular solution is unlikely. However, much research in condensed matter physics is directed at systems with quenched-in randomness, in which the atoms are not all alike. An important feature of such systems, termed "frustration," is that interactions favoring different and incompatible kinds of ordering may be simultaneously present (*9*). The magnetic alloys known as "spin glasses," which exhibit competition between ferromagnetic and antiferromagnetic spin ordering, are the best understood example of frustration (*10*). It is now believed that highly frustrated systems like spin glasses have many nearly degenerate random ground states rather than a single ground state with a high degree of symmetry. These systems stand in the same relation to conventional magnets as glasses do to crystals, hence the name.

The physical properties of spin glasses at low temperatures provide a possible guide for understanding the possibilities of optimizing complex systems subject to conflicting (frustrating) constraints.

Physical Design of Computers

The physical design of electronic systems and the methods and simplifications employed to automate this process have been reviewed (*11, 12*). We first provide some background and definitions related to applications of the simulated annealing framework to specific problems that arise in optimal design of computer systems and subsystems. Physical design follows logical design. After the detailed specification of the logic of a system is complete, it is necessary to specify the precise physical realization of the system in a particular technology.

This process is usually divided into several stages. First, the design must be partitioned into groups small enough to fit the available packages, for example, into groups of circuits small enough to fit into a single chip, or into groups of chips and associated discrete components that can fit onto a card or other higher level package. Second, the circuits are assigned specific locations on the chip. This stage is usually called placement. Finally, the circuits are connected by wires formed photolithographically out of a thin metal film, often in several layers. Assigning paths, or routes, to the wires is usually done in two stages. In rough or global wiring, the wires are assigned to regions that represent schematically the capacity of the intended package. In detailed wiring (also called exact embedding), each wire is given a unique complete path. From the detailed wiring results, masks can be generated and chips made.

At each stage of design one wants to optimize the eventual performance of the system without compromising the feasibility of the subsequent design stages. Thus partitioning must be done in such a way that the number of circuits in each partition is small enough to fit easily into the available package, yet the number of signals that must cross partition boundaries (each requiring slow, power-consuming driver circuitry) is minimized. The major focus in placement is on minimizing the length of connections, since this translates into the time required for propagation of signals, and thus into the speed of the finished system. However, the placements with the shortest implied wire lengths may not be wirable, because of the presence of regions in which the wiring is too congested for the packaging technology. Congestion, therefore, should also be anticipated and minimized during the placement process. In wiring, it is desirable to maintain the minimum possible wire lengths while minimizing sources of noise, such as cross talk between adjacent wires. We show in this and the next two sections how these conflicting goals can be combined and made the basis of an automatic optimization procedure.

The tight schedules involved present major obstacles to automation and optimization of large system design, even when computers are employed to speed up the mechanical tasks and reduce the chance of error. Possibilities of feedback, in which early stages of a design are redone to solve problems that became apparent only at later stages, are greatly reduced as the scale of the overall system being designed increases. Op-

a(x,y) = min(1, exp{(E(x)-E(y))/T})
T: large — > accept bad state
T: small — > reject bad state

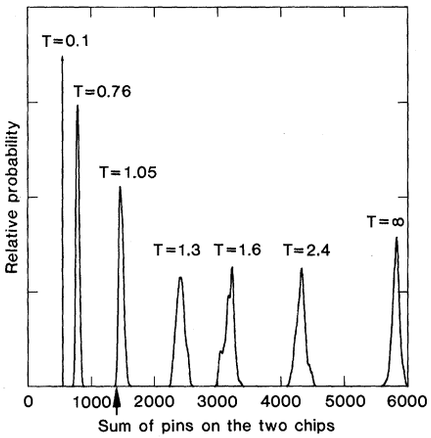


Fig. 1. Distribution of total number of pins required in two-way partition of a microprocessor at various temperatures. Arrow indicates best solution obtained by rapid quenching as opposed to annealing.

timization procedures that can incorporate, even approximately, information about the chance of success of later stages of such complex designs will be increasingly valuable in the limit of very large scale.

System performance is almost always achieved at the expense of design convenience. The partitioning problem provides a clean example of this. Consider N circuits that are to be partitioned between two chips. Propagating a signal across a chip boundary is always slow, so the number of signals required to cross between the two must be minimized. Putting all the circuits on one chip eliminates signal crossings, but usually there is no room. Instead, for later convenience, it is desirable to divide the circuits about equally.

If we have connectivity information in a matrix whose elements $\{a_{ij}\}$ are the number of signals passing between circuits i and j, and we indicate which chip circuit i is placed on by a two-valued variable $\mu_i = \pm 1$, then N_c, the number of signals that must cross a chip boundary is given by $\Sigma_{i>j}(a_{ij}/4)(\mu_i - \mu_j)^2$. Calculating $\Sigma_i \mu_i$ gives the difference between the numbers of circuits on the two chips. Squaring this imbalance and introducing a coefficient, λ, to express the relative costs of imbalance and boundary crossings, we obtain an objective function, f, for the partition problem:

$$f = \sum_{i>j} \left(\lambda - \frac{a_{ij}}{2}\right)\mu_i\mu_j \qquad (7)$$

Reasonable values of λ should satisfy $\lambda \lesssim z/2$, where z is the average number of circuits connected to a typical circuit (fan-in plus fan-out). Choosing $\lambda \simeq z/2$ implies giving equal weight to changes in the balance and crossing scores.

The objective function f has precisely the form of a Hamiltonian, or energy function, studied in the theory of random magnets, when the common simplifying assumption is made that the spins, μ_i, have only two allowed orientations (up or down), as in the linear chain example of the previous section. It combines local, random, attractive ("ferromagnetic") interactions, resulting from the a_{ij}'s, with a long-range repulsive ("antiferromagnetic") interaction due to λ. No configuration of the $\{\mu_i\}$ can simultaneously satisfy all the interactions, so the system is "frustrated," in the sense formalized by Toulouse (*9*).

If the a_{ij} are completely uncorrelated, it can be shown (*13*) that this Hamiltonian has a spin glass phase at low temperatures. This implies for the associated magnetic problem that there are many degenerate "ground states" of nearly equal energy and no obvious symmetry. The magnetic state of a spin glass is very stable at low temperatures (*14*), so the ground states have energies well below the energies of the random high-temperature states, and transforming one ground state into another will usually require considerable rearrangement. Thus this analogy has several implications for optimization of partition:

1) Even in the presence of frustration, significant improvements over a random starting partition are possible.

2) There will be many good near-optimal solutions, so a stochastic search procedure such as simulated annealing should find some.

3) No one of the ground states is significantly better than the others, so it is not very fruitful to search for the absolute optimum.

In developing Eq. 7 we made several severe simplifications, considering only two-way partitioning and ignoring the fact that most signals connect more than two circuits. Objective functions analogous to f that include both complications are easily constructed. They no longer have the simple quadratic form of Eq. 7, but the qualitative feature, frustration, remains dominant. The form of the Hamiltonian makes no difference in the Metropolis Monte Carlo algorithm. Evaluation of the change in function when a circuit is shifted to a new chip remains rapid as the definition of f becomes more complicated.

It is likely that the a_{ij} are somewhat correlated, since any design has considerable logical structure. Efforts to understand the nature of this structure by analyzing the surface-to-volume ratio of components of electronic systems [as in "Rent's rule" (*15*)] conclude that the

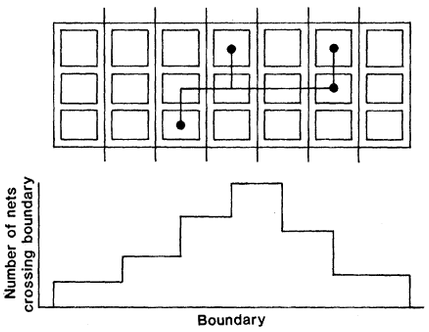


Fig. 2. Construction of a horizontal net-crossing histogram.

circuits in a typical system could be connected with short-range interactions if they were embedded in a space with dimension between two and three. Uncorrelated connections, by contrast, can be thought of as infinite-dimensional, since they are never short-range.

The identification of Eq. 7 as a spin glass Hamiltonian is not affected by the reduction to a two- or three-dimensional problem, as long as $\lambda N \simeq z/2$. The degree of ground state degeneracy increases with decreasing dimensionality. For the uncorrelated model, there are typically of order $N^{1/2}$ nearly degenerate ground states (*14*), while in two and three dimensions, $2^{\alpha N}$, for some small value, α, are expected (*16*). This implies that finding a near-optimum solution should become easier, the lower the effective dimensionality of the problem. The entropy, measurable as shown in Eq. 6, provides a measure of the degeneracy of solutions. $S(T)$ is the logarithm of the number of solutions equal to or better than the average result encountered at temperature T.

As an example of the partitioning problem, we have taken the logic design for a single-chip IBM "370 microprocessor" (*17*) and considered partitioning it into two chips. The original design has approximately 5000 primitive logic gates and 200 external signals (the chip has 200 logic pins). The results of this study are plotted in Fig. 1. If one randomly assigns gates to the two chips, one finds the distribution marked $T = \infty$ for the number of pins required. Each of the two chips (with about 2500 circuits) would need 3000 pins. The other distributions in Fig. 1 show the results of simulated annealing.

Monte Carlo annealing is simple to implement in this case. Each proposed configuration change simply flips a randomly chosen circuit from one chip to the other. The new number of external connections, C, to the two chips is calculated (an external connection is a net with circuits on both chips, or a circuit

connected to one of the pins of the original single-chip design), as is the new balance score, B, calculated as in deriving Eq. 7. The objective function analogous to Eq. 7 is

$$f = C + \lambda B \tag{8}$$

where C is the sum of the number of external connections on the two chips and B is the balance score. For this example, $\lambda = 0.01$.

For the annealing schedule we chose to start at a high "temperature," $T_0 = 10$, where essentially all proposed circuit flips are accepted, then cool exponentially, $T_n = (T_1/T_0)^n T_0$, with the ratio $T_1/T_0 = 0.9$. At each temperature enough flips are attempted that either there are ten accepted flips per circuit on the average (for this case, 50,000 accepted flips at each temperature), or the number of attempts exceeds 100 times the number of circuits before ten flips per circuit have been accepted. If the desired number of acceptances is not achieved at three successive temperatures, the system is considered "frozen" and annealing stops.

The finite temperature curves in Fig. 1 show the distribution of pins per chip for the configurations sampled at $T = 2.5$, 1.0, and 0.1. As one would expect from the statistical mechanical analog, the distribution shifts to fewer pins and sharpens as the temperature is decreased. The sharpening is one consequence of the decrease in the number of configurations that contribute to the equilibrium ensemble at the lower temperature. In the language of statistical mechanics, the entropy of the system decreases. For this sample run in the low-temperature limit, the two chips required 353 and 321 pins, respectively. There are 237 nets connecting the two chips (requiring a pin on each chip) in addition to the 200 inputs and outputs of the original chip. The final partition in this example has the circuits exactly evenly distributed between the two partitions. Using a more complicated balance score, which did not penalize imbalance of less than 100 circuits, we found partitions resulting in chips with 271 and 183 pins.

If, instead of slowly cooling, one were to start from a random partition and accept only flips that reduce the objective function (equivalent to setting $T = 0$ in the Metropolis rule), the result is chips with approximately 700 pins (several such runs led to results with 677 to 730 pins). Rapid cooling results in a system frozen into a metastable state far from the optimal configuration. The best result obtained after several rapid quenches is indicated by the arrow in Fig. 1.

Placement

Placement is a further refinement of the logic partitioning process, in which the circuits are given physical positions (*11, 12, 18, 19*). In principle, the two stages could be combined, although this is not often possible in practice. The objectives in placement are to minimize

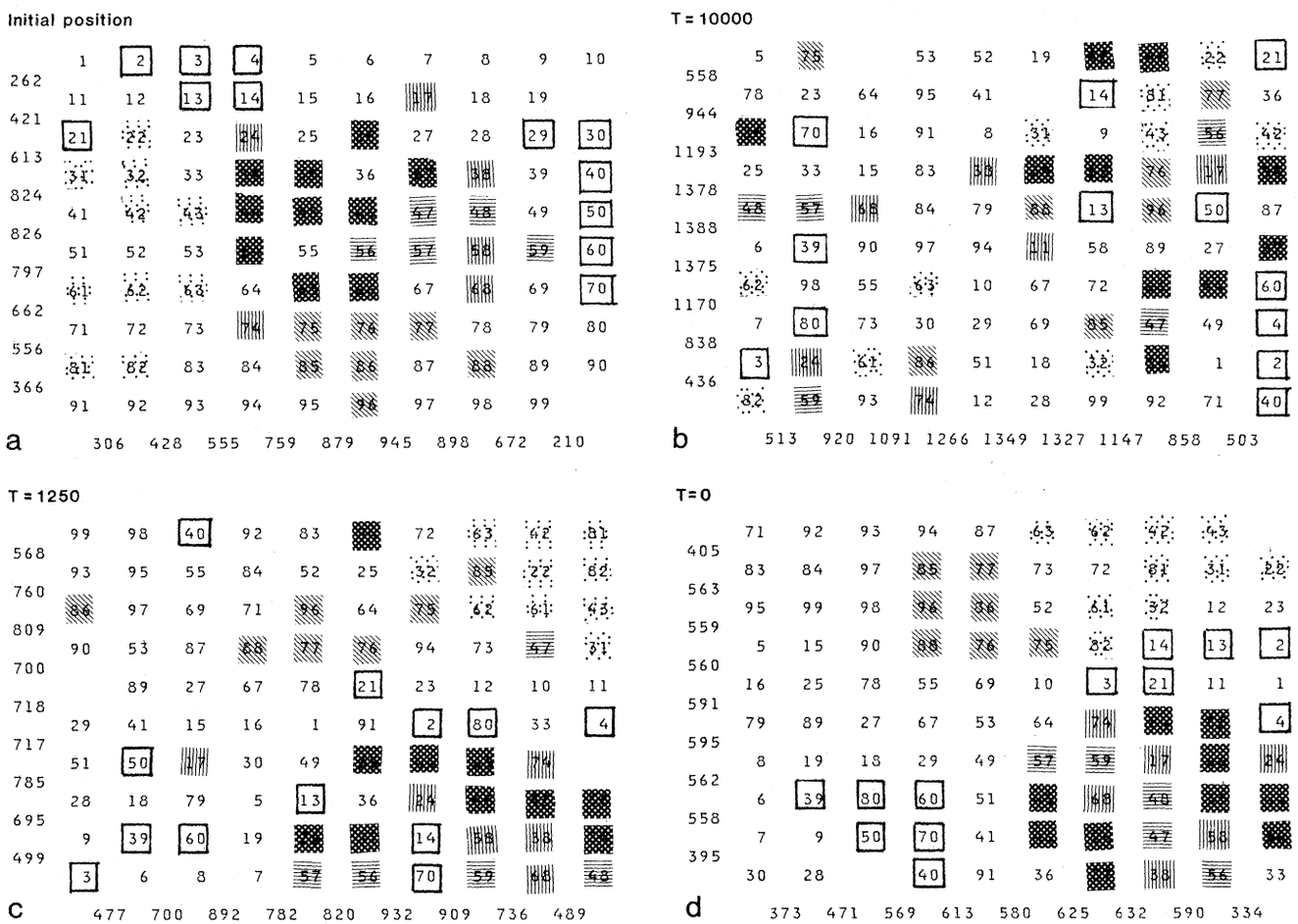


Fig. 3. Ninety-eight chips on a ceramic module from the IBM 3081. Chips are identified by number (1 to 100, with 20 and 100 absent) and function. The dark squares comprise an adder, the three types of squares with ruled lines are chips that control and supply data to the adder, the lightly dotted chips perform logical arithmetic (bitwise AND, OR, and so on), and the open squares denote general-purpose registers, which serve both arithmetic units. The numbers at the left and lower edges of the module image are the vertical and horizontal net-crossing histograms, respectively. (a) Original chip placement; (b) a configuration at $T = 10{,}000$; (c) $T = 1250$; (d) a zero-temperature result.

signal propagation times or distances while satisfying prescribed electrical constraints, without creating regions so congested that there will not be room later to connect the circuits with actual wire.

Physical design of computers includes several distinct categories of placement problems, depending on the packages involved (*20*). The larger objects to be placed include chips that must reside in a higher level package, such as a printed circuit card or fired ceramic "module" (*21*). These chip carriers must in turn be placed on a backplane or "board," which is simply a very large printed circuit card. The chips seen today contain from tens to tens of thousands of logic circuits, and each chip carrier or board will provide from one to ten thousand interconnections. The partition and placement problems decouple poorly in this situation, since the choice of which chip should carry a given piece of logic will be influenced by the position of that chip.

The simplest placement problems arise in designing chips with structured layout rules. These are called "gate array" or "master slice" chips. In these chips, standard logic circuits, such as three- or four-input NOR's, are preplaced in a regular grid arrangement, and the designer specifies only the signal wiring, which occupies the final, highest, layers of the chip. The circuits may all be identical, or they may be described in terms of a few standard groupings of two or more adjacent cells.

As an example of a placement problem with realistic complexity without too many complications arising from package idiosyncrasies, we consider 98 chips packaged on one multilayer ceramic module of the IBM 3081 processor (*21*). Each chip can be placed on any of 100 sites, in a 10 × 10 grid on the top surface of the module. Information about the connections to be made through the signal-carrying planes of the module is contained in a "netlist," which groups sets of pins that see the same signal.

The state of the system can be briefly represented by a list of the 98 chips with their x and y coordinates, or a list of the contents of each of the 100 legal locations. A sufficient set of moves to use for annealing is interchanges of the contents of two locations. This results in either the interchange of two chips or the interchange of a chip and a vacancy. For more efficient search at low temperatures, it is helpful to allow restrictions on the distance across which an interchange may occur.

To measure congestion at the same time as wire length, we use a convenient intermediate analysis of the layout, a net-crossing histogram. Its construction is summarized in Fig. 2. We divide the package surface by a set of natural boundaries. In this example, we use the boundaries between adjacent rows or columns of chip sites. The histogram then contains the number of nets crossing each boundary. Since at least one wire must be routed across each boundary crossed, the sum of the entries in the histogram of Fig. 2 is the sum of the horizontal extents of the rectangles bounding each net, and is a lower bound to the horizontal wire length required. Constructing a vertical net-crossing histogram and summing its entries gives a similar estimate of the vertical wire length.

The peak of the histogram provides a lower bound to the amount of wire that must be provided in the worst case, since each net requires at least one wiring channel somewhere on the boundary. To combine this information into a single objective function, we introduce a threshold level for each histogram—an amount of wire that will nearly exhaust the available wire capacity—and then sum for all histogram elements that exceed the threshold the square of the excess over threshold. Adding this quantity to the estimated length gives the objective function that was used.

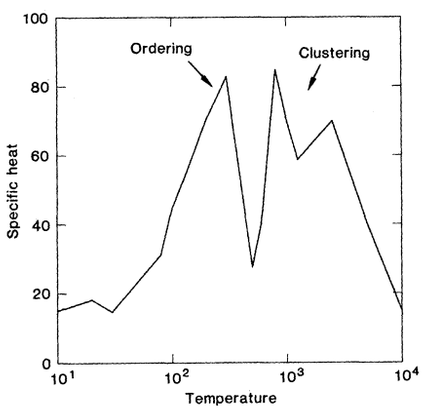


Fig. 4. Specific heat as a function of temperature for the design of Fig. 3, a to d.

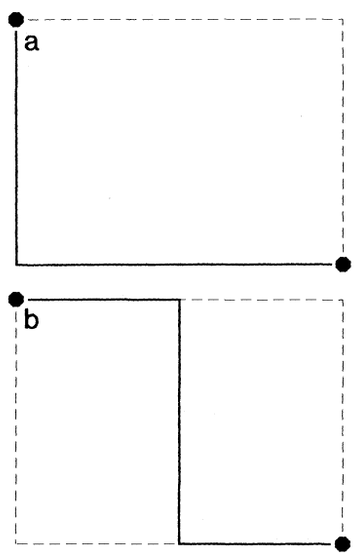


Fig. 5. Examples of (a) L-shaped and (b) Z-shaped wire rearrangements.

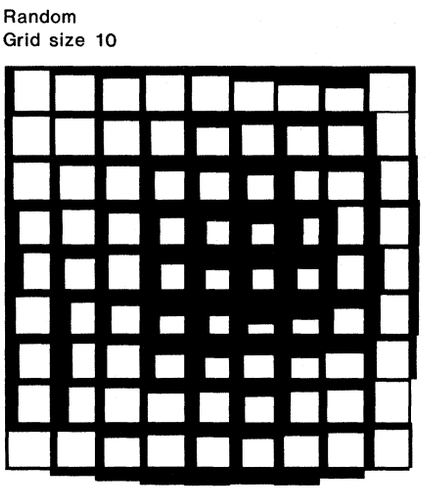
Figure 3 shows the stages of a simulated annealing run on the 98-chip module. Figure 3a shows the chip locations from the original design, with vertical and horizontal net-crossing histograms indicated. The different shading patterns distinguish the groups of chips that carry out different functions. Each such group was designed and placed together, usually by a single designer. The net-crossing histograms show that the center of the layout is much more congested than the edges, most likely because the chips known to have the most critical timing constraints were placed in the center of the module to allow the greatest number of other chips to be close to them.

Heating the original design until the chips diffuse about freely quickly produces a random-looking arrangement, Fig. 3b. Cooling very slowly until the chips move sluggishly and the objective function ceases to decrease rapidly with change of temperature produced the result in Fig. 3c. The net-crossing histograms have peaks comparable to the peak heights in the original placement, but are much flatter. At this "freezing point," we find that the functionally related groups of chips have reorganized from the melt, but now are spatially separated in an overall arrangement quite different from the original placement. In the final result, Fig. 3d, the histogram peaks are about 30 percent less than in the original placement. Integrating them, we find that total wire length, estimated in this way, is decreased by about 10 percent. The computing requirements for this example were modest: 250,000 interchanges were attempted, requiring 12 minutes of computation on an IBM 3033.

Between the temperature at which clusters form and freezing starts (Fig. 3c) and the final result (Fig. 3d) there are many further local rearrangements. The functional groups have remained in the same regions, but their shapes and relative alignments continue to change throughout the low-temperature part of the annealing process. This illustrates that the introduction of temperature to the optimization process permits a controlled, adaptive division of the problem

through the evolution of natural clusters at the freezing temperature. Early prescription of natural clusters is also a central feature of several sophisticated placement programs used in master slice chip placement (*22, 23*).

A quantity corresponding to the thermodynamic specific heat is defined for this problem by taking the derivative with respect to temperature of the average value of the objective function observed at a given temperature. This is plotted in Fig. 4. Just as a maximum in the specific heat of a fluid indicates the onset of freezing or the formation of clusters, we find specific heat maxima at two temperatures, each indicating a different type of ordering in the problem. The higher temperature peak corresponds to the aggregation of clusters of functionally related objects, driven apart by the congestion term in the scoring. The lower temperature peak indicates the further decrease in wire length obtained by local rearrangements. This sort of measurement can be useful in practice as a means of determining the temperature ranges in which the important rearrangements in the design are occurring, where slower cooling with be helpful.



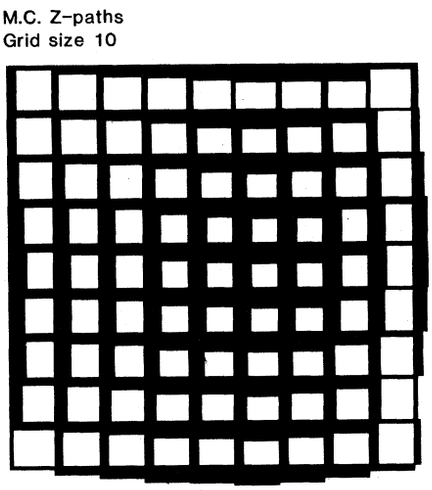


Fig. 6 (left). Wire density in the 98-chip module with the connections randomly assigned to perimeter routes. Chips are in the original placement. Fig. 7 (right). Wire density after simulated annealing of the wire routing, using Z-shaped moves.

Wiring

After placement, specific legal routings must be found for the wires needed to connect the circuits. The techniques typically applied to generate such routings are sequential in nature, treating one wire at a time with incomplete information about the positions and effects of the other wires (*11, 24*). Annealing is inherently free of this sequence dependence. In this section we describe a simulated annealing approach to wiring, using the ceramic module of the last section as an example.

Nets with many pins must first be broken into connections—pairs of pins joined by a single continuous wire. This "ordering" of each net is highly dependent on the nature of the circuits being connected and the package technology. Orderings permitting more than two pins to be connected are sometimes allowed, but will not be discussed here.

The usual procedure, given an ordering, is first to construct a coarse-scale routing for each connection from which the ultimate detailed wiring can be completed. Package technologies and structured image chips have prearranged areas of fixed capacity for the wires. For the rough routing to be successful, it must not call for wire densities that exceed this capacity.

We can model the rough routing problem (and even simple cases of detailed embedding) by lumping all actual pin positions into a regular grid of points, which are treated as the sources and sinks of all connections. The wires are then to be routed along the links that connect adjacent grid points.

The objectives in global routing are to minimize wire length and, often, the number of bends in wires, while spreading the wire as evenly as possible to simplify exact embedding and later revision. Wires are to be routed around regions in which wire demand exceeds capacity if possible, so that they will not "overflow," requiring drastic rearrangements of the other wires during exact embedding. Wire bends are costly in packages that confine the north-south and east-west wires to different layers, since each bend requires a connection between two layers. Two classes of moves that maintain the minimum wire length are shown in Fig. 5. In the L-shaped move of Fig. 5a, only the essential bends are permitted, while the Z-shaped move of Fig. 5b introduces one extra bend. We will explore the optimization possible with these two moves.

For a simple objective function that will reward the most balanced arrangement of wire, we calculate the square of the number of wires on each link of the network, sum the squares for all links, and term the result F. If there are N_L links and N_W wires, a global routing program that deals with a high density of wires will attempt to route precisely the average number of wires, N_W/N_L, along each link. In this limit F is bounded below by N_W^2/N_L. One can use the same objective function for a low-density (or high-resolution) limit appropriate for detailed wiring. In that case, all the links have either one or no wires, and links with two or more wires are illegal. For this limit the best possible value of F will be N_W/N_L.

For the L-shaped moves, F has a relatively simple form. Let $\epsilon_{iv} = +1$ along the links that connection i has for one orientation, -1 for the other orientation, and 0 otherwise. Let a_{iv} be 1 if the ith connection can run through the vth link in either of its two positions, and 0 otherwise. Note that a_{iv} is just ϵ_{iv}^2. Then if $\mu_i = \pm 1$ indicates which route the ith connection has taken, we obtain for the number of wires along the vth link,

$$n_v = \sum_i \frac{a_{iv}(\epsilon_{iv}\mu_i + 1)}{2} + n_v(0) \qquad (9)$$

where $n_v(0)$ is the contribution from straight wires, which cannot move without increasing their length, or blockages.

Summing the n_v^2 gives

$$F = \sum_{i,j} J_{ij}\mu_i\mu_j + \sum_i h_i\mu_i + \text{constants} \qquad (10)$$

which has the form of the Hamiltonian for a random magnetic alloy or spin glass, like that discussed earlier. The "random field," h_i, felt by each movable connection reflects the difference, on the average, between the congestion associated with the two possible paths:

$$h_i = \sum_v \epsilon_{iv}\,[2n_v(0) + \sum_j a_{jv}] \qquad (11)$$

The interaction between two wires is proportional to the number of links on which the two nets can overlap, its sign depending on their orientation conventions:

$$J_{ij} = \sum_v \frac{\epsilon_{iv}\epsilon_{jv}}{4} \qquad (12)$$

Both J_{ij} and h_i vanish, on average, so it is the fluctuations in the terms that make up F which will control the nature of the low-energy states. This is also true in spin glasses. We have not tried to exhibit a functional form for the objective function with Z-moves allowed, but simply calculate it by first constructing the actual amounts of wire found along each link.

To assess the value of annealing in wiring this model, we studied an ensemble of randomly situated connections, under various statistical assumptions. Here we consider routing wires for the 98 chips on a module considered earlier. First, we show in Fig. 6 the arrangement of wire that results from assigning each wire to an L-shaped path, choosing orientations at random. The thickness of the links is proportional to the number of wires on each link. The congested area that gave rise to the peaks in the histograms discussed above is seen in the wiring just below and to the right of the center of the module. The maximum numbers of wires along a single link in Fig. 6 are 173 (x direction) and 143 (y direction), so the design is also anisotropic. Various ways of rearranging the wiring paths were studied. Monte Carlo annealing with Z-moves gave the best solution, shown in Fig. 7. In this example, the largest numbers of wires on a single link are 105 (x) and 96 (y).

We compare the various methods of improving the wire arrangement by plotting (Fig. 8) the highest wire density found in each column of x-links for each of the methods. The unevenness of the density profiles was already seen when we considered net-crossing histograms as input information to direct placement. The lines shown represent random assignment of wires with L-moves; aligning wires in the direction of least average congestion—that is, along h_i—followed by cooling for one pass at zero T; simulated annealing with L-moves only; and annealing with Z-moves. Finally, the light dashed line shows the optimum result, in which the wires are distributed with all links carrying as close to the average weight as possible. The optimum cannot be attained in this example without stretching wires beyond their minimum length, because the connections are too unevenly arranged. Any method of optimization gives a significant improvement over the estimate obtained by assigning wire routings at random. All reduce the peak wire density on a link by more than 45 percent. Simulated annealing with Z-moves improved the random routing by 57 percent, averaging results for both x and y links.

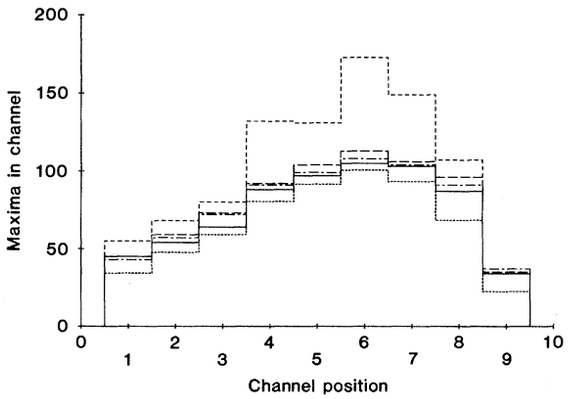


Fig. 8. Histogram of the maximum wire densities within a given column of x-links, for the various methods of routing.

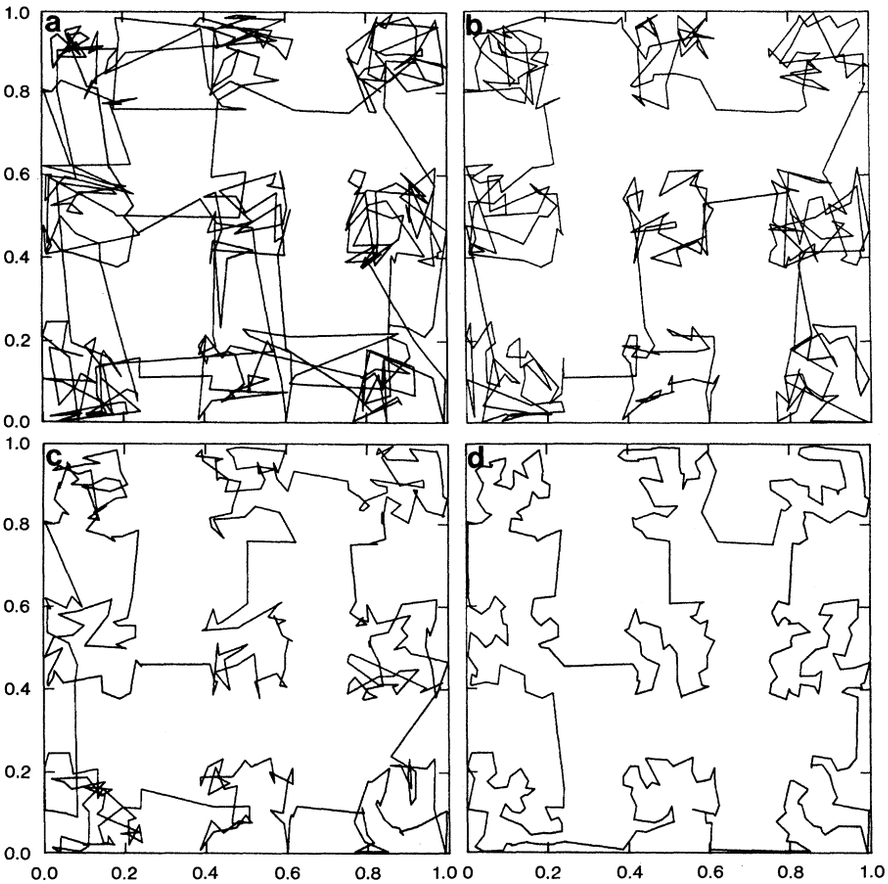


Fig. 9. Results at four temperatures for a clustered 400-city traveling salesman problem. The points are uniformly distributed in nine regions. (a) $T = 1.2$, $\alpha = 2.0567$; (b) $T = 0.8$, $\alpha = 1.515$; (c) $T = 0.4$, $\alpha = 1.055$; (d) $T = 0.0$, $\alpha = 0.7839$.

Traveling Salesmen

Quantitative analysis of the simulated annealing algorithm or comparison between it and other heuristics requires problems simpler than physical design of computers. There is an extensive literature on algorithms for the traveling salesman problem (*3, 4*), so it provides a natural context for this discussion.

If the cost of travel between two cities is proportional to the distance between them, then each instance of a traveling salesman problem is simply a list of the positions of N cities. For example, an arrangement of N points positioned at random in a square generates one instance. The distance can be calculated in either the Euclidean metric or a "Manhattan" metric, in which the distance between two points is the sum of their separations along the two coordinate axes. The latter is appropriate for physical design applications, and easier to compute, so we will adopt it.

We let the side of the square have length $N^{1/2}$, so that the average distance between each city and its nearest neighbor is independent of N. It can be shown that this choice of length units leaves the optimal tour length per step independent of N, when one averages over many

instances, keeping N fixed (*25*). Call this average optimal step length α. To bound α from above, a numerical experiment was performed with the following "greedy" heuristic algorithm. From each city, go to the nearest city not already on the tour. From the Nth city, return directly to the first. In the worst case, the ratio of the length of such a greedy tour to the optimal tour is proportional to $\ln(N)$ (*26*), but on average, we find that its step length is about 1.12. The variance of the greedy step length decreases as $N^{-1/2}$, so the situation envisioned in the worst case analysis is unobservably rare for large N.

To construct a simulated annealing algorithm, we need a means of representing the tour and a means of generating random rearrangements of the tour. Each tour can be described by a permuted list of the numbers 1 to N, which represents the cities. A powerful and general set of moves was introduced by Lin and Kernighan (*27, 28*). Each move consists of reversing the direction in which a section of the tour is traversed. More complicated moves have been used to enhance the searching effectiveness of iterative improvement. We find with the adaptive divide-and-conquer effect of annealing at intermediate temperatures that the subsequence reversal moves are sufficient (*29*).

An annealing schedule was determined empirically. The temperature at which segments flow about freely will be of order $N^{1/2}$, since that is the average bond length when the tour is highly random. Temperatures less than 1 should be cold. We were able to anneal into locally optimal solutions with $\alpha \leq 0.95$ for N up to 6000 sites. The largest traveling salesman problem in the plane for which a proved exact solution has been obtained and published (to our knowledge) has 318 points (*30*).

Real cities are not uniformly distributed, but are clumped, with dense and sparse regions. To introduce this feature into an ensemble of traveling salesman problems, albeit in an exaggerated form, we confine the randomly distributed cities to nine distinct regions with empty gaps between them. The temperature gives the simulated annealing method a means of separating out the problem of the coarse structure of the tour from the local details. At temperatures, such as $T = 1.2$ (Fig. 9a), where the small-scale structure of the paths is completely disordered, the longer steps across the gaps are already becoming infrequent and steps joining regions more than one gap are eliminated. The configurations studied below $T = 0.8$ (for instance, Fig. 9b) had the minimal number of long steps, but the detailed arrangement of the long steps continued to change down to $T = 0.4$ (Fig. 9c). Below $T = 0.4$, no further changes in the arrangement of the long steps were seen, but the small-scale structure within each region continued to evolve, with the result shown in Fig. 9d.

Summary and Conclusions

Implementing the appropriate Metropolis algorithm to simulate annealing of a combinatorial optimization problem is straightforward, and easily extended to new problems. Four ingredients are needed: a concise description of a configuration of the system; a random generator of "moves" or rearrangements of the elements in a configuration; a quantitative objective function containing the trade-offs that have to be made; and an annealing schedule of the temperatures and length of times for which the system is to be evolved. The annealing schedule may be developed by trial and error for a given problem, or may consist of just warming the system until it is obviously melted, then cooling in slow stages until diffusion of the components ceases. Inventing the most effective sets of moves and deciding which factors to incorporate into the objective function require insight into the problem being solved and may not be obvious. However, existing methods of iterative improvement can provide natural elements on which to base a simulated annealing algorithm.

The connection with statistical mechanics offers some novel perspectives on familiar optimization problems. Mean field theory for the ordered state at low temperatures may be of use in estimating the average results to be obtained by optimization. The comparison with models of disordered interacting systems gives insight into the ease or difficulty of finding heuristic solutions of the associated optimization problems, and provides a classification more discriminating than the blanket "worst-case" assignment of many optimization problems to the NP-complete category. It appears that for the large optimization problems that arise in current engineering practice a "most probable" or average behavior analysis will be more useful in assessing the value of a heuristic than the traditional worst-case arguments. For such analysis to be useful and accurate, better knowledge of the appropriate ensembles is required.

Freezing, at the temperatures where large clusters form, sets a limit on the energies reachable by a rapidly cooled spin glass. Further energy lowering is possible only by slow annealing. We expect similar freezing effects to limit the effectiveness of the common device of employing iterative improvement repeatedly from different random starting configurations.

Simulated annealing extends two of the most widely used heuristic techniques. The temperature distinguishes classes of rearrangements, so that rearrangements causing large changes in the objective function occur at high temperatures, while the small changes are deferred until low temperatures. This is an adaptive form of the divide-and-conquer approach. Like most iterative improvement schemes, the Metropolis algorithm proceeds in small steps from one configuration to the next, but the temperature keeps the algorithm from getting stuck by permitting uphill moves. Our numerical studies suggest that results of good quality are obtained with annealing schedules in which the amount of computational effort scales as N or as a small power of N. The slow increase of effort with increasing N and the generality of the method give promise that simulated annealing will be a very widely applicable heuristic optimization technique.

Dunham (*5*) has described iterative improvement as the natural framework for heuristic design, calling it "design by natural selection." [See Lin (*6*) for a fuller discussion.] In simulated annealing, we appear to have found a richer framework for the construction of heuristic algorithms, since the extra control provided by introducing a temperature allows us to separate out problems on different scales.

Simulation of the process of arriving at an optimal design by annealing under control of a schedule is an example of an evolutionary process modeled accurately by purely stochastic means. In fact, it may be a better model of selection processes in nature than is iterative improvement. Also, it provides an intriguing instance of "artificial intelligence," in which the computer has arrived almost uninstructed at a solution that might have been thought to require the intervention of human intelligence.

References and Notes

1. E. L. Lawlor, *Combinatorial Optimization* (Holt, Rinehart & Winston, New York, 1976).
2. A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, Mass., 1974).
3. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).
4. R. Karp, *Math. Oper. Res.* **2**, 209 (1977).
5. B. Dunham, *Synthese* **15**, 254 (1963).
6. S. Lin, *Networks* **5**, 33 (1975).
7. For a concise and elegant presentation of the basic ideas of statistical mechanics, see E. Shrödinger, *Statistical Thermodynamics* (Cambridge Univ. Press, London, 1946).
8. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
9. G. Toulouse, *Commun. Phys.* **2**, 115 (1977).