Ensemble approach to simulated annealing
George Ruppeiner, Jacob Mørch Pedersen, Peter Salamon

To cite this version:

HAL Id: jpa-00246344
https://hal.science/jpa-00246344
Submitted on 4 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Ensemble approach to simulated annealing

George Ruppeiner (1), Jacob Mørch Pedersen (2, *) and Peter Salamon (3)

(1) Division of Natural Sciences, New College of the University of South Florida, Sarasota, Florida 34243, U.S.A.
(2) Physics Laboratory, H. C. Ørsted Institute, University of Copenhagen, Universitetsparken 5, DK-2100 Copenhagen, Denmark
(3) Department of Mathematical Sciences, San Diego State University, San Diego, California 92182, U.S.A.

(Received 16 November 1990, accepted 9 January 1991)

Abstract. — We present three reasons for implementing simulated annealing with an ensemble of random walkers which search the configuration space in parallel. First, an ensemble allows the implementation of an adaptive cooling schedule because it provides good statistics for collecting thermodynamic information. This new adaptive implementation is general, simple, and, in some sense, optimal. Second, the ensemble can tell us how to optimally allocate our effort in the search for a good solution, i.e., given the total computer time available, how many members to use in the ensemble. Third, an ensemble can reveal otherwise hidden properties of the configuration space, e.g., by examining Hamming distance distributions among the ensemble members. We present numerical results on the bipartitioning of random graphs and on a graph bipartitioning problem whose static thermodynamic properties may be solved for exactly.

1. Introduction.

Simulated annealing is a class of computer algorithms for finding good solutions to certain computationally intractable problems [1]. It is based on an analogy with statistical mechanics, particularly the statistical mechanics of spin glasses [1, 2]. The objective is to minimize the cost functions for problems which have many local minima in their space of configurations.

Most previous approaches to simulated annealing used a serial search repeated several times to get a good solution. Among the few parallel studies, the typical procedure [3] divides the configuration space into cells. In view of the phenomenon of broken ergodicity, this seems unnecessary, since the accessible configuration space will in any case fragment into cells [4]. Better to let the nature of the space decide the division! We advocate a parallel search of the entire configuration space simultaneously by an ensemble of random walkers with limited communication. Specifically, we allow the random walkers to send information giving the

(*) Permanent address: Ødegaard & Danneskiold-Samsøe ApS., Krogsgade 1, Copenhagen, Denmark.
values of the energies they visited to a central control routine which decides when to update the shared value of the temperature.

We advocate a parallel search of the entire configuration space for three reasons. First, it enables us to get good statistics for thermodynamic quantities with ensemble averages. Good statistics make the implementation of our adaptive schedule possible.

Second, it enables us to compute collective properties, such as Hamming distances, among the members of the ensemble. Hamming distances provide insight into the nature of the configuration space of the problem. In particular, they illuminate the distribution of local energy minima in these very complicated spaces.

Third, the ensemble makes it possible to calculate the distribution of Best So Far Energies, (BSFE), as a function of time [5]. The BSFE is the minimum energy a walker has seen up to time t. The BSFE distribution for all walkers allows a determination of the optimal ensemble size for a given computational effort which minimizes the expectation value of the lowest energy seen.

The idea of ensembles is clearly dictated by the analogy to real physical systems. While our arguments do not rely on this directly, the intuition for them is often motivated by the analogy. We strongly believe that the analogy to physical systems is useful and runs far deeper than what is presently understood. Simulated annealing should be judged with an implementation which makes full use of the intuition gained from the analogy rather than merely the idea of «slow cooling».

Let us add that an essential aspect concerning our approach to simulated annealing is «broken ergodicity», which is of fundamental importance in spin glass problems [4]. At low temperatures, time averages are not equal to ensemble averages even after a very long time since the system gets stuck in regions of phase space from which escape is, practically speaking, impossible. This makes the ensemble approach fundamentally different in character from the serial approach.

2. Theory.

In this section, we discuss the theory behind our method. The discussion divides naturally into four sections: 2.1 background, 2.2 implementation, 2.3 Best So Far Energies (BSFE), and 2.4 Hamming distances.

2.1 BACKGROUND. — Let us introduce the example we use to illustrate the ideas in this paper — the problem of graph bipartitioning [2]. A graph consists of a collection of vertices connected by edges [6]. The problem is to divide the vertices into two equal size sets, A and B, such that the number of edges running between the sets is minimized. Each way of assigning vertices to sets is a «configuration» and the «cost function» for any configuration is the number of edges connecting vertices in different subsets.

A practical application of graph bipartitioning is provided by circuit design, where the vertices are components of an electronic circuit and the edges are electrical connections between the components. How should the components be divided equally between two chips to minimize the number of connections between the chips?

The difficulty in graph bipartitioning is that it appears one cannot, for a general graph, be guaranteed of having found the optimal solution, i.e., the configuration with the smallest value of the cost function, unless essentially all of the configurations have been tried, and the number of possible configurations grows exponentially with the number of vertices. Graph bipartitioning is an NP-complete problem, for which no efficient algorithm for finding the optimal solution is known [7]. The objective, therefore, is to find efficient algorithms for finding good solutions.
A graph consists of vertices, and of edges connecting some pairs of vertices. The vertices are numbered from 1 to \(N\), where we take \(N\) to be an even number. Two vertices connected by an edge are said to be adjacent.

There are

\[
\binom{N}{\frac{N}{2}} = \frac{N!}{\left(\frac{N}{2}\right)! \left(\frac{N}{2}\right)!} = 2^N
\]

possible configurations for the partition of a graph into two equal size subsets A and B. For \(N = 52\), it would take a computer which can test a million configurations per second about sixteen years to test every configuration. The problem with 54 vertices would take about four times as long. Typical problem sizes in circuit design [1] have \(N = 5000\). It is clear that the most we can hope to find is a good configuration.

An analogy is made between the cost function in simulated annealing and the energy in statistical mechanics. Henceforth, we call the cost function the energy. Configurations in simulated annealing are analogous to microstates in statistical mechanics.

To approach the graph bipartitioning problem, imagine that there is a «walker» constrained to travel over the configuration space looking for the optimal solution. A clock keeps time. As the time increases by one unit, the walker considers making a «step» from one configuration to another. A step corresponds to switching a randomly selected vertex in subset A with a randomly selected vertex in subset B. An attempted step is accepted with the probability

\[
P(\Delta E, T) = \begin{cases} 
1 & \text{if } \Delta E \leq 0 \\
\exp(-\Delta E/T) & \text{if } \Delta E > 0 
\end{cases}
\]

given by the Metropolis algorithm [1, 8]. \(\Delta E\) is the change in energy which would result from accepting the step, and \(T\) is a parameter called the temperature. If \(T\) is infinite, all possible steps are accepted, and the search is entirely random. If \(T\) is very small, uphill moves are essentially forbidden, and the search is a descent search, or «quench», which ceases to accept further attempted moves once the walker reaches a local minimum. The idea in simulated annealing is to start at a high temperature and cool slowly so that the walker is continually drawn towards low energy configurations, while at the same time being able to escape local minima along the way.

Our step rule for the random walk in configuration space defines a dynamics for the problem. For fixed \(T\), it leads to the Gibbs-Boltzmann distribution

\[
p(\omega, T) = \frac{\exp[-E(\omega)/T]}{Z(T)},
\]

in the limit of infinite time, regardless of the initial configuration [1, 8]. Here, \(p(\omega, T)\) is the probability of getting the configuration \(\omega\), with energy \(E(\omega)\), at temperature \(T\) and

\[
Z(T) = \sum_{\omega} \exp[-E(\omega)/T]
\]

is the partition function. The sum is over all possible configurations.

The equilibrium average energy \(\langle E \rangle_T\) at some temperature \(T\) is

\[
\langle E \rangle_T = \sum_{\omega} E(\omega) p(\omega, T).
\]
The equilibrium heat capacity is given by a fluctuation formula:

\[ C(T) = \frac{d \langle E \rangle_T}{dT} = \frac{\langle (\Delta E)^2 \rangle_T}{T^2}, \quad (6) \]

where

\[ \langle (\Delta E)^2 \rangle_T = \sum_{\omega} [E(\omega) - \langle E \rangle_T]^2 p(\omega, T). \quad (7) \]

In our approach to simulated annealing, we use an ensemble of walkers to calculate thermodynamic averages. The ensemble of walkers continually tries to reach equilibrium with a «reservoir» at temperature \( T \), which is the temperature in the Metropolis probability, equation (2). We refer to this reservoir as the «target». A key assumption in the justification of optimization, but not in the implementation of our method, is that the energies of the ensemble of walkers are at all times distributed according to the Gibbs-Boltzmann distribution at some temperature.

The two examples of graph partitioning problems employed in the present paper are random graphs and a simple family of graphs called «necklaces». Random graphs are graphs in which each possible edge is present or not with a certain probability. An instance of the graph is generated by using this property to decide independently whether each edge is present. Necklaces are a family of graphs whose thermodynamic properties can be solved for exactly [9]. An \([m, n]\) necklace consists of a cycle of \( m \) elements, each element is a completely connected graph with \( n \) vertices. In this paper, we have restricted \( m \) to be even, and used \( n = 2 \). Figure 1 shows the \([8, 2]\) necklace in a configuration of lowest energy. There are \( m \) lowest energy configurations related by «rotations». In our dynamics, each optimal configuration is separated from the one next to it by an energy barrier of height 2.
Bipartitioning necklaces is not an NP-complete problem. Nonetheless, it possesses features, such as many local minima and broken ergodicity, which lead to insight into more difficult problems. In a similar spirit, Ettelaie and Moore have used a one dimensional Ising spin glass to study simulated annealing [10].

A fundamental difficulty with simulated annealing for graph bipartitioning is that there are regions of the configuration space which are dynamically inaccessible from one another on the time scale of observation, particularly at low temperatures. This causes ergodicity to break down over time scales even much longer than those characteristic of simulated annealing runs [4]. Hence, time averages taken for a single walker are not the same as ensemble averages taken over the entire configuration space. It follows that the ensemble approach is quite different from the serial approach. Thus the ensemble approach is well worth exploring even aside from the question of computational speed.

If we cool starting with an ensemble in equilibrium at some temperature, then the ensemble members must rearrange themselves within relatively isolated regions of the configuration space, and they must also rearrange between the regions. While the first rearrangement is likely to be relatively fast, the latter likely takes considerable time and results in the ensemble failing to reach equilibrium. The only way to alleviate this is to cool very slowly. Our general approach, based on ensemble equilibrium for justification of optimality, may be viewed as a possible improvement, but it also suffers from very long equilibration times at low temperatures.

2.2 Schedule. — A critical issue is how to best carry out simulated annealing runs. What is the best sequence of temperatures, and how much time should be spent at each temperature? Most previous applications of simulated annealing have used preset schedules based either on simplicity of implementation or, in the case of the Geman and Geman schedule [11, 12], on the fact that it has provable asymptotic properties. Here we present an adaptive schedule whose implementation depends on the use of ensembles.

Let \( t \) denote the « time » measured as the number of Metropolis steps attempted per walker. The Geman and Geman schedule uses \( T(t) = d/\ln (1 + t) \), where \( d \) is a constant. For sufficiently large \( d \), this schedule results in a random walk which, in the limit of infinite time, visits the ground state with probability 1. However, no one has used a value of \( d \) which gives a sufficiently slow cooling rate for the proven asymptotic properties to apply.

More typically, simulated annealing studies have used a constant cooling rate \( T(t) = a - bt \) [10, 13-14] or an exponential cooling rate \( T(t) = a \exp (- bt) \) [15], where \( a \) and \( b \) are constants. Our interest is in a certain schedule which makes use of thermodynamic information gathered during the cooling. Several cooling schedules have been advanced which make use of thermodynamic criteria. Examples of such schedules include ones which keep \( \Delta E \) [16, 17] or \( \Delta S \) [18] constant for each relaxation from \( T(t) \) to \( T(t + 1) \). Such schedules can improve performance as compared with exponential or linear cooling. Our constant thermodynamic speed schedule keeps the average energy of the system within a fixed number of standard deviations of the energy which the system would reach in equilibrium [5, 19-24]. The constant thermodynamic speed schedule has been shown to be optimal in a certain sense specified below and has performed very well in simulations [5, 20-23].

Let \( \langle E \rangle \) be the average energy of the ensemble of walkers at some time, and let \( \langle E \rangle_T \) be the average energy, as given by equation (5), corresponding to \( T \). (Note, though, that equation (5) is never actually used to calculate \( \langle E \rangle_T \)!) In our schedule, \( T \) is controlled so as to keep \( \langle E \rangle_T \) a fixed number \( v \) of standard deviations less than \( \langle E \rangle \):

\[
\langle E \rangle_T = \langle E \rangle - v \langle (\Delta E)^2 \rangle^{1/2},
\]  
(8)
where \( v \) is the «thermodynamic speed» which is kept constant, and 
\[
\langle (\Delta E)^2 \rangle = \langle (E - \langle E \rangle)^2 \rangle
\]
is the variance of the energies of the ensemble members. The idea
is to evaluate \( \langle E \rangle \) often, and adjust \( \langle E \rangle_T \) to maintain the equality in equation (8).

Following each calculation of a new \( \langle E \rangle_T \) using equation (8), a new \( T \) must be determined
for calculating the step probabilities. We refer to \( \langle E \rangle_T \) and \( T \) as the target energy and
temperature, respectively. Let \( d \langle E \rangle_T \) and \( dT \) be the change, from the old values, of the target
energy and the target temperature, respectively. From equation (6),

\[
dT = T^2 \frac{d \langle E \rangle_T}{\langle (\Delta E)^2 \rangle}
\]

Details of our implementation will be discussed further in the computation section where we
present an explicit algorithm.

Our schedule is consistent with the common wisdom that cooling should slow down if either
the heat capacity [1] or the time scale [25] gets large.

We tested our implementation of simulated annealing against some other cooling schedules
[23]. Our implementation was found to produce generally superior results.

Our schedule is optimal in two senses. The first is based on the constraint that the state of
the ensemble should remain at all times statistically «indistinguishable» from the equilibrium
state toward which the system is striving [19, 26]. The second is the fact that our schedule
minimizes entropy production in the thermodynamic analog of a system cooled to a low
temperature by successive contact with cooler and cooler reservoirs [23, 27].

2.3 BSF ENERGY. — The Best So Far Energy of the \( i \)-th walker at time \( t \), BSFE\(_i\)(\( t \)), is
defined as the lowest energy that walker has seen up to time \( t \):

\[
\text{BSFE}_i(t) = \min_{0 \leq t' \leq t} \{ E_i(t') \}
\]

where \( E_i(t') \) is the energy of the \( i \)-th walker at time \( t' \).

We have used BSFE's as a measure of quality in comparing different optimization methods
[5, 23, 28, 29]. Here we show how one can use BSFE's to determine the optimal ensemble size
from an annealing run, for given total number of Metropolis steps, or computational effort
\( c = M^* t \), where \( M \) is the number of walkers [30].

Denote the distribution of BSFE's by \( f(E, t) \). From \( f(E, t) \) it is possible to calculate the
distribution of the Very Best So Far Energy, VBSFE, which is the minimum energy seen up to
time \( t \) over the entire ensemble:

\[
\text{VBSFE}(t) = \min_{0 \leq t \leq M} \text{BSFE}_i(t).
\]

Denote the probability distribution of VBSFE's by \( h(E, t, M) \). This distribution is found from
its cumulative distribution \( H(E, t, M) \) which is given by:

\[
H(E, t, M) = 1 - \left( \frac{1}{1 - \int_{-\infty}^{E} f(E', t) dE'} \right)^M
\]

The integral in equation (12) is the probability that the BSFE is less than or equal to \( E \). One
minus this integral is then the probability that the BSFE is greater than \( E \). This difference to
the \( M \)-th power is the probability that all \( M \) independent samples give a BSFE of \( E \) or greater.
$H(E, t, M)$ is therefore the probability that at least one of the $M$ walkers has visited a state with energy $E$ or lower.

The probability density $h(E, t, M)$ is:

$$h(E, t, M) = \frac{\partial H(E, t, M)}{\partial E},$$

with expectation value

$$\langle \text{VBSFE}(t, M) \rangle = \int_{-\infty}^{\infty} Eh(E, t) \, dE.$$  \tag{14}

For a given total computer effort $c$, it is possible to minimize $\langle \text{VBSFE}(t, M) \rangle$ by varying the ensemble size $M$. At least one local minimum for $\langle \text{VBSFE}(t, M) \rangle$ should exist since for very small $M$ the walkers sample too little of the configuration space to find a low energy, and for very large $M$ too little time is allocated to each walker to get very far. We let $M^* = M^*(c)$ denote the optimal ensemble size for fixed $c$.

Using an ensemble in simulated annealing allows us to sample and hence estimate the distribution $f(E, t)$. From this distribution we can find the optimal ensemble size $M^*$ a posteriori. As yet, no run-time method for choosing $M^*$ has been found.

To get a good estimate of the distribution $f(E, t)$ it is necessary to run the annealing with an ensemble. The calculation can serve as a guideline for choosing an ensemble size $M$ on similar problems.

2.4 Hamming Distances. — A measure of the nature of the configuration space being sampled by the ensemble at any instant is provided by the distribution of Hamming distances between the walkers. Each graph vertex contributes either 0 or 1 to the Hamming distance between two walkers; the contribution is 0 if the vertex is in the same subset for both walkers and 1 otherwise. The sum of the contributions of all the vertices is the Hamming distance between the two walkers. The Hamming distance is a natural measure for distance in configuration space.

There are $M(M-1)/2$ distinct pairs of walkers. We note the following facts: (1) the Hamming distance $H$ between two walkers lies between 0 and the number of vertices $N$, inclusive; (2) $H$ is an even number; (3) in the limit as $M$ gets large, the Hamming distance distribution is symmetric about $N/2$; (4) at high temperatures, where the walkers are randomly distributed over the configuration space, the probability distribution $P_N(H)$ of Hamming distances $H$ is

$$P_N(H) = \begin{cases} 0 & \text{if } H \text{ odd} \\ 2^{1-N} \frac{N!}{(N-H)!H!} & \text{if } H \text{ even} \end{cases}$$  \tag{15}

Condition (3) holds since the configuration space is symmetric under a swap of all A and B vertices. If such a swap is made for a walker, the Hamming distance with any other walker goes from $H$ to $N - H$. This expected symmetry offers a good runtime check as to whether or not enough walkers are being used.

As the temperature is lowered, walkers should tend to group together in low energy regions of the configuration space and the Hamming distribution should spread out from a central peak. The distribution of Hamming distances is a direct measure of the effective configuration space currently being sampled by the ensemble.

In problems with a unique optimal configuration, the Hamming distance distribution should approach two spikes at 0 and $N$ as the temperature is slowly reduced to zero.
Fig. 2. — Flowchart of our main program. The expression for New $E$ comes from equation (8) and the one for New $T$ from equation (9). The averages are the ensemble averages. A program constant $t_r$ contained the number of times each walker was considered at each temperature. In our runs, this constant was set at 100.

Also illuminating are the Hamming distributions for the walkers with a specific energy. This allows viewing the configuration space at a particular energy slice, for example, the lowest energy configurations.

3. Computer program.

The major portion of our program was written in MacFORTH. For speed, the often repeated parts were written in 68 000 ASSEMBLY language. Thirty-one bit random numbers were generated with Tautworth's algorithm (R250) which is fast, and has an essentially infinite cycle time [31, 32]. Below we describe the data structures and algorithms used by our program.

The graph data structure consists of three parts: first, a constant giving the total number of vertices, second, a list of neighbors for each vertex [33], and, third, a table of $N$ addresses, one address for each vertex neighbor list. This data structure was picked to minimize the time required to read off all the neighbors of any vertex, and to minimize the amount of memory required to store the graph. An adjacency matrix type representation of $N \times N$ flags, in
which the $i, j$-th matrix element is 1 if vertices $i$ and $j$ are connected, and 0 otherwise, takes up much more memory for sparse graphs, and is slower reading off all the neighbors of a vertex. On the other hand, with an adjacency matrix representation, given any two vertices, one may look up directly whether or not they are neighbors. This must be known to compute energy changes. With our data structure for the graph, this question required a list search.

The list of neighbors for each vertex began with the number of neighbors of the vertex, followed by a list of the neighbors. The neighbor vertex numbers were listed in ascending order to allow use of the efficient binary search algorithm.

The walkers were numbered from 1 to $M$, where $M$ is a constant. Each walker data structure consists of two parts: the energy of the walker and an array of $N$ flags which gives the subset to which each vertex belongs. If two vertices are switched, both walker vertex subset list and walker energy are updated.

The target data structure consists of three parts: the target temperature $T$, the target energy $\langle E \rangle_T$, and a table of Metropolis probabilities corresponding to the target temperature, one element for every possible $\Delta E$. The probabilities were stored as scaled up integers to speed up the arithmetic.

We kept track of the BSFE's in an array with one element for each walker. Following each update of a walker, the walker's energy was compared with his corresponding BSFE. If the walker's energy was less than his BSFE, then his current energy replaced his BSFE.

In addition there were constants for the thermodynamic speed $v$ and for the number of times each walker was considered before updating the target. There was also a constant controlling how frequently analysis was saved to a text file for later use. A counter was used to keep track of the total number of times each walker was considered during a run.

The program proceeded very simply, as shown in figure 2. During initialization, all data structures were prepared. With a new graph, the initial temperature was picked fairly high by the user, and the walkers were allowed to equilibrate at that temperature. This allowed the determination of the first target energy. The criterion for choosing the first temperature was that the Hamming distance distribution should approximately match the high temperature distribution in equation (15). Then each walker in the ensemble was considered several times (100 times for all the runs reported here) for a Metropolis step at the temperature $T$. After this, a new target was set and the process continued until the average energy ceased to terminate.

4. Results.

In this section, we present results of our computer experiments with several 100 vertex random graphs, and with the [80, 2] necklace.

We begin with the necklace results since these are easy to interpret. We made a sequence of runs at various thermodynamic speeds on the [80, 2] necklace. Each run started with an ensemble of 500 walkers in equilibrium at $T = 2.0$. Equilibrium was achieved by running at this temperature for several thousand steps per walker. This temperature is well above $T = 0.42$, the location of the peak in the heat capacity. The Hamming distance distribution $T = 2.0$ is reasonably close to that of the high temperature distribution in equation (15). During the cooling, the target was updated after each 100 attempted steps per walker.

Figure 3a shows the target energy $\langle E \rangle_T$ as a function of the target temperature $T$ for several speeds. Also shown is the theoretical curve [9]. The points for all these speeds are close to one another, and to the theoretical curve, indicating that the temperature did not change too fast at any of the speeds shown. Ultimately, several walkers found an optimal configuration of the [80, 2] necklace after each of the runs.
Fig. 3. — Target energy $\langle E \rangle_T$ for the [80, 2] necklace as a function of the target temperature $T$ computed along the way for four speeds. Also shown is the theoretical curve for $\langle E \rangle_T$. Figure 3b is a magnification of a section of the graph at low temperatures. At low temperatures, the experimental curve falls above the theoretical one for all speeds. We believe that this is an indication of non-equilibrium brought on by long relaxation times. The horizontal arrow in figure 3a shows where the theoretical curve crosses the left edge of the graph.

Figure 3b shows a magnification of the graph at low temperature. It is clear that the points fall closer to the theoretical curve for smaller speeds. However, the limiting curve is approached slowly with decreasing speed, indicating the presence of very long relaxation times. The ensemble method offers a simple limiting technique to at least test whether or not we are approaching equilibrium along the way. Note that despite the presence of very long relaxation times, the experimental results do not deviate excessively from the theoretical results.

Fig. 4. — Ensemble energy standard deviation as a function of the ensemble average energy for $v = 0.05$. The scatter is less than 10% about the theoretical curve. Good data for $\langle (\Delta E)^2 \rangle^{1/2}$ are essential for the implementation of our adaptive schedule.

Essential ingredients in the implementation of our schedule are good run time estimates of $\langle (\Delta E)^2 \rangle^{1/2}$. Figure 4 shows $\langle (\Delta E)^2 \rangle^{1/2}$ for the ensemble of 500 walkers as a function of the average ensemble energy $\langle E \rangle$. Also shown is the theoretical equilibrium curve. Both the quality of the data and the agreement with theory are good.
Fig. 5. — Target temperature $T$ as a function of the Metropolis time $t$ for the [80, 2] necklace for three values of the thermodynamic speed. All of the runs started with an ensemble of 500 walkers in equilibrium at $T = 2.0$. For these three speeds, the lowest average energy was eventually achieved with $v = 0.2$. The runs with $v = 0.5$ got stuck in local energy minima too soon, and the run with $v = 0.05$ was too slow to reach really low temperatures in the time allotted.

Figure 5 shows the temperature schedule followed with several speeds. Cooling is rapid at first, but slows as the temperature decreases and relaxation times grow.

We turn now to our results on the random graph. Consider first random graph A which consists of $N = 100$ vertices, and was constructed with probability of any pair of vertices being connected $p = 0.05$. Random graph A was used in reference [23], where it was shown that the constant thermodynamic speed schedule produced a lower BSF Energy distribution than several other schedules encountered in the literature. This reference also showed a graph of the heat capacity for random graph A which has a well defined peak at $T = 0.87 \pm 0.03$. The constant thermodynamic speed schedule proved to follow roughly the same cooling pattern as a modified Geman and Geman schedule [12, 34]. We note, however, that the Geman and Geman schedule is not adaptive, and has parameters which may be difficult to select.

Fig. 6. — Target energy $\langle E \rangle_T$ as a function of the target temperature $T$ for Random Graph A, which has 100 vertices and $p = 0.05$, for three speeds. Figure 6b is a magnification of a section of the graph at low temperatures.
We made a sequence of constant thermodynamic speed coolings on random graph A. Each cooling used an ensemble of 500 walkers starting in equilibrium at $T = 2.0$ and ran for a total time of about 100 000 steps per walker.

Figure 6 shows the target energy as a function of the target temperature for several speeds. Qualitatively, the results look similar to those in figure 3 for the [80, 2] necklace graph. We have no exact theoretical solution to compare with the experimental results for bipartitioning a random graph.

Figure 7 shows $\langle (\Delta E)^2 \rangle^{1/2}$ as a function of $\langle E \rangle$ taken in the run with $v = 0.1$. Again the results look qualitatively similar to the corresponding ones for the [80, 2] necklace. The scatter with 500 walkers is not excessive.

The cooling schedules followed for random graph A are shown in reference [23]. The characteristic cooling pattern evident is roughly the same as that for the [80, 2] necklace: initially rapid cooling, followed by slower cooling.

Of considerable interest is the nature of the configuration spaces for problems such as those considered here. Palmer [35] has posed a set of questions regarding the nature of the configuration space, some of which we take up here. Perhaps most important are the nature and the distribution of the deep energy minima. Are the minimum energy configurations at the bottom of broad valleys, or are they more like narrow holes in a flat golf course? Are the sides of the valleys reasonably smooth, or are there numerous ledges and traps to impede a search for the bottom? Are the locations of the deep energy minima correlated? In other words, are the optimal configurations close to one another in the configuration space, or are they far apart? These questions are amenable to analysis using ensembles.

We consider first Random Graph A discussed above. In the course of the run with $v = 0.1$, we analyzed the Hamming distance distributions of the walkers with specific energies. Figure 8 shows the Hamming distance distribution for this graph for four energy slices at and above what we believe to be the lowest energy $E = 47$. It is clear that there is a multiplicity of states with lowest energy $E = 47$. In addition, they seem to be clustered about two regions of the phase space.

A trivial degeneracy in the lowest energy configuration is due to vertices which are not connected to any other vertex. Swapping such vertices results in $\Delta E = 0$. Random Graph A has only one unconnected vertex, so this is not the cause of this degeneracy. The degeneracy results because of reasonably far separated valleys in configuration space.
Fig. 8. — Hamming distance distribution for Random Graph A for walkers with four specific energies. For each graph, the number of walkers at each energy is given. Also show for each graph is the temperature along the way at which the data was computed. The scale of the frequency distribution is different for each graph.

Fig. 9. — Hamming distance distributions for eight random graphs. Shown are the distributions for the lowest and the next to the lowest energies for each graph.
We point out that not only was it relatively easy to find what we believed to be the lowest energy state \( (E = 47) \) with many walkers, but on a straight quench to \( T = 0 \) about 20\% of the walkers found a lowest energy configuration. This indicates that the sides of the large valleys are reasonably smooth with enough byways to slide down even on a straight quench.

We examined nine random graphs with \( N = 100 \) constructed with connection probability \( p = 0.05 \). For all of these graphs it was easy to find what we believe to be the lowest energy states with many walkers. The average lowest energy for these graphs is \( 52 \pm 3 \), which compares with 42 predicted by Fu and Andersen [2] and 44 by Banaver et al. [36]. Considering the approximations made in these theoretical calculations, we consider this to be good agreement. These theoretical predictions were based on series expansions for sparse graphs; our random graphs were not all that sparse.

Figure 9 shows that for most graphs the lowest energy states tend to be few in number and clustered closely together in one region of the configuration space. The next to highest energy shows more structure as the bottoms of additional valleys make their appearance.

Our general conclusion about the random graphs we studied is that the optimal energies are to be found at the bottoms of rather broad valleys with reasonably smooth walls. In more cases than not, the lowest energy configurations were in a single small region of the configuration space.

In general, we conclude that our ensemble approach offers an excellent way to explore the configuration space of complicated systems by means of the Hamming distance distribution for the ensemble of walkers.

Figure 10 shows the expectation value of the very lowest energy observed over the ensemble as a function of the ensemble size \( M \). The total computer effort is calculated as the product of the ensemble size and the time allocated to each walker and was kept constant. The annealed graph is a 200 vertex random graph with connection probability of 0.01. The annealing was done with an ensemble of 200 walkers with \( v = 0.1 \).

5. Conclusions.

We have presented a new implementation of simulated annealing. In contrast to most previous implementations, which are based on repetitive serial searches, our approach is based on an ensemble: many copies searching simultaneously over the entire configuration space.

Using an ensemble, one can get run-time estimates of collective properties which are useful in at least three respects. First, we showed how to use the first two moments of the current
ensemble energy distribution to implement an adaptive annealing schedule \( T(t) \) which is in some sense optimal. Second, we showed how the ensemble BSFE’s (Best So Far Energies) can be used to select the ensemble size which allocates computational effort optimally in the sense of finding the lowest expectation value of the energy of the best configuration seen by any member of the ensemble. Third, we showed how the ensemble Hamming distance distribution can reveal interesting information concerning the structure of the state space of the problem. The symmetry of the Hamming distance distribution, or lack thereof, can indicate whether our ensemble size is sufficiently large for accurate statistics.

The spirit of simulated annealing is simplicity and generality; the algorithm should not require much knowledge about the problem to which it is applied. Our approach does not depart from this spirit.

The statistical mechanical approach has much to offer at this general level where the algorithm uses very little about the structure of the specific problem. Going to ensembles is a natural step suggested by the analogy to statistical mechanics. We believe this analogy runs deep; our results show that it can be exploited further.

We tested our algorithm on bipartitioning of random graphs as well as of a class of graphs for which one can get exact thermodynamic information.

We acknowledge James D. Nulton for useful conversations, and the Telluride Summer Research Center for providing a stimulating environment. One of us, J. P., acknowledges the Danish Natural Research Council for partial support.

References

[6] We work with finite, undirected graphs without loops or multiple edges; see, e.g., Harary, F., Graph Theory (Addison-Wesley, Reading, Mass., 1969).
[33] We call two vertices neighbors if they are connected by an edge.