Chapter 11

Genetic Algoritms

11.1 Hill Climbing

If we know that the objective function has only one maximum then all up-hill trails lead to that maximum and we can find its location (and therefore the best-fitting model) by simply stepping uphill again and again from any convenient starting point. There are many ways of going up the surface of the objective function to a maximum; they generally rely on estimates of the derivatives of the objective function in the neighborhood of the starting point. These techniques are collectively called "hill-climbing" and are usually quite straightforward: follow the gradient vector uphill until the gradient is zero or as close to zero are you are willing to tolerate. If there is only one local maximum then the resulting model is **the** best solution to the inverse problem.

Unfortunately, many geophysically interesting inverse problems have objective functions which are highly complicated and possess many local maxima. As a consequence, the result of an iterative, gradient-following, or hill-climbing calculation may depend strongly on the starting model: the algorithm may very well converge to a local maximum which is not the globally highest peak.

An example of this complexity is shown in Figure 11.1. The computation from which this was taken is a synthetic model of surface-consistent reflection statics with 55 model parameters (50 statics delays and 5 auxiliary parameters) and is due to Christof Stork. The figure shows the cross-correlation-based "goodness" function as a function of two of the variable model parameters when the remaining 53 were set to their exactly-correct values. (The central and largest peak is the global maximum for this problem.) The figure suggests that the final answer we would get from a "hill-climbing" method will depend strongly on where the climb begins (that is, on our choice of starting model). If we chose a starting point at random from the domain shown in Figure 11.1, hill climbing would almost always lead to an inferior solution (we would not climb the highest peak).

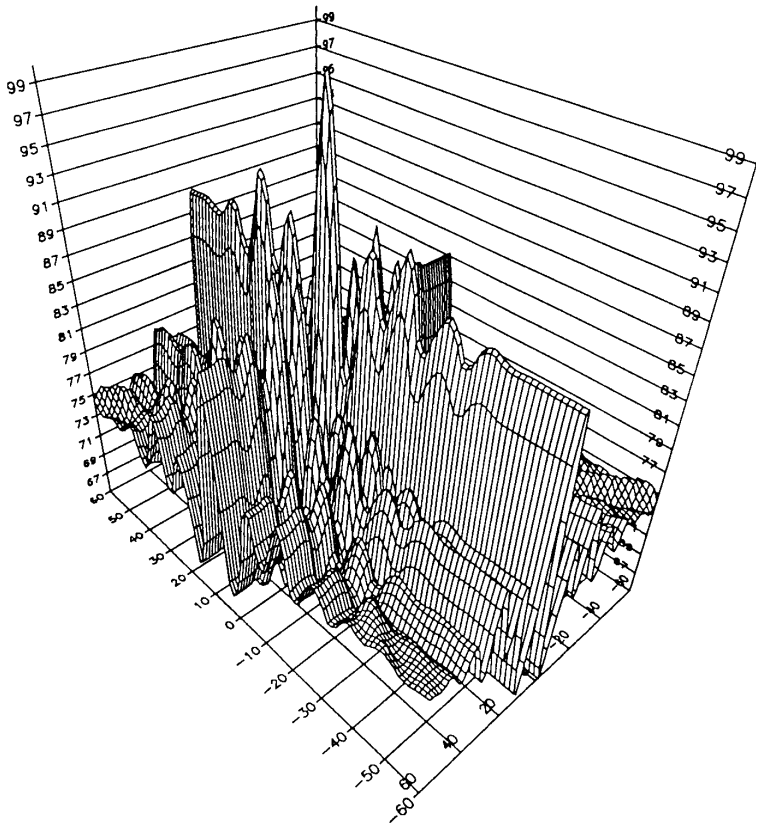

Figure 11.1: The "goodness" function for a model surface-consistent statics calculation involving 55 model statics parameters; the greater the value of this function, the better the model's parameters remove statics shifts in the data set (which is not shown). For this figure 53 of the 55 parameters were set to the correct value and the remaining two were varied.

Hill-climbing methods are *local* optimization methods. At each step the algorithm only examines the immediate vicinity of the current point in order to choose its next step; the algorithm makes no attempt to reconnoiter distant portions of model space. This limitation has two consequences for optimization:

- A local method only explores the single peak upon whose flanks it began.
- Local methods can be very efficient in reaching the hill's top.

Where local methods are *appropriate*, they can be made to perform very well indeed. Where local methods are *inappropriate*, they are likely to efficiently climb the wrong hill.

11.2 Global Optimization

In contrast to local methods, *global* methods are optimization techniques which make some explicit effort to search more widely in model space. Thhree such methods are:

- exhaustive search,
- simulated annealing, and
- genetic algorithms.

We will very briefly mention a few properties of the first two of these classes, and then devote the remainder of this note to a discussion of the third class: genetic algorithms.

11.3 Exhaustive Search

This straightforward scheme consists of simply evaluating the objective function for every distinct model and reporting the model that produces the largest value. The only effort involved in formulating such a calculation is deciding what constitutes "distinctness" between models. All global methods require some such decision, and it usually amounts to just converting continuous parameters to quantized ones.

Exhaustive search has the interesting properties that it is very simple to implement, it is guaranteed to work, and it is almost always useless in practice. The problem with exhaustive search in almost every case of interest is that the searcher becomes exhausted long before the model space. For example, in the reflection statics example above (Figure 11.1) the model space consists of 55 parameters each of which was allowed to range over

about 40 values. This fairly simple example has a model space with $40^{55} \approx 10^{88}$ distinct models. If we could evaluate 10^9 models per second (one *gigamod per second*?), it would take us about 3×10^{72} years to search all of model space.

11.4 Simulated Annealing.

This class of techniques is based upon a close analogy between optimization and the growth of long-range order (such as large crystals) in a slowly-cooling melt. Simulated anealing has been the subject of much interest since it was invented in 1983 by Kirkpatrick, Gelatt, and Vecchi [KGV83]. In the last two years, major advances have been made in reliability and efficiency.

Simulated annealing, which will be discussed in detail in the next chapter, is usually implemented as a sort of biased "drunkard's walk" in model space. The drunkard's path through model space is the result of a competition between two tendencies: one is to walk uphill to the nearest peak, and the other is to take a step in a randomly chosen direction. When the drunkard is really plastered, practically all of his steps are chosen at random; when the drunkard is relatively sober, practically all of his steps are uphill. During the course of a simulated annealing calculation, the drunkard's state is varied slowly from plastered to sober. The algorithm functions as a kind of hill-climbing in which the climber is able to change hill from time to time.

Simulated annealing is a kind of quasi-local algorithm: it relies on sampling the space of models by following along a path from nearest neighbor to nearest neighbor. As a result, the computational complexity of the method is dominated by a grid size or step length that is essentially unrelated to the underlying structure of the objective function. This leads to a phenomenon known as "critical slowing down" or "the curse of dimensionality" as the size of the problem grows.

11.5 Genetic Algorithms

This "new" class of global optimization algorithms, often referred to as GA's, actually predate simulated annealing by more than a decade. The concepts involved were discovered and developed in the late 1960s and early 1970s by John Holland and his students. Holland's book [Hol75] describes the theory behind these algorithms. The field is quite active; see [Dav91], [Gre87], [Sch89], [BB91]. (Also note the useful introductory text by Goldberg [Gol89].)

Simulated annealing is based upon a close analogy between function optimization and a physical system composed of particles that interact in a relatively simple way but are

enormous in number. Genetic algorithms proceed from a loose analogy between function optimization and a biological system composed of organisms that interact in a relatively complex way and are comparatively few in number. A genetic algorithm tries to find an optimal answer by evolving a population of trial answers in a way that mimics biological evolution. If simulated annealing "cooks" an answer, then genetic algorithms "breed" one.

11.5.1 Inside a GA

To be a little more specific, suppose that we have a scalar valued function which is defined over some domain. Since our mental model is that of a geophysical inverse problem, we refer to points in the domain as "models" and the value of the function at some model as that model's "fitness." Our goal is to find a model, or possibly a set of several models, that maximizes the fitness function: we want to find the best ("fittest") models in the space of all possible models. (In practice we'll often settle for finding some "pretty fit" models.)

The essence of a GA is that it's a set of operations which we apply to a *population* of models that lead us to a new population in such a way that the members of the new population will have a greater expected average fitness than their predecessors. Suppose that we are executing a genetic algorithm that has a constant population size of 15 (there are many variations and this is a plausible one). Then at some point in the calculation, we have a set of 15 models and we have computed the fitness for each of these. Figure (11.2) is a cartoon showing a population with 15 members for an application in which a model consists of a single real number, x, in the interval $0 \leq x \leq 1$. Also shown is the fitness of each model, as a vertical line from the abscissa to the curve of the fitness function. These models were selected at random (which is how GA's are usually started); Figure (11.5), to which we shall turn later, shows the population when the search has progressed farther.

We then do our "GA-thing" on the population, and are led to a new population of models (many of which may be identical to members of the predecessor population). The transition processes which tell us how to evolve the new population from the old one are designed so that (among other things) the *expected average fitness* of the new population is at least as large as the average fitness of the old one. We simply apply this processes repetitively until we have found a satisfactory answer (or we have given up).

The details of these transition processes (which are the heart of GA's) vary widely, but all share a basic tripartite structure. The elements of that structure are

- **selection**: designating which members of the current population will be allowed an opportunity to pass on their characteristics to the next generation.
- **recombination**: contructing new child models by combining model features copied

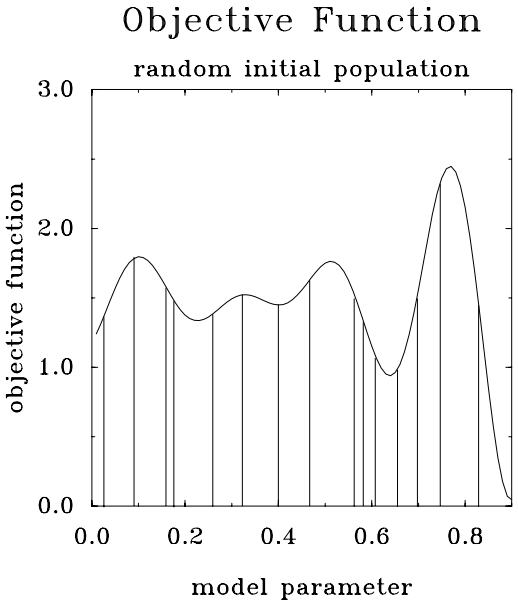

Figure 11.2: A schematic display of a model population in an application where a model is specified by a single real number in the range [0,1]. The population consists of fifteen randomly-chosen points from model space. Also shown is the objective (or "fitness") function as well as the value achieved by each model in the population.

from the set of selected parent models.

- **mutation**: randomly perturbing the model parameters of an occasional child model (for the purpose of adding diversity to the population).

The literature cited earlier, like the field itself, abounds with a lush and bewildering variety of genetic algorithms. We must content ourselves here with describing the particular algorithm we actually used (which is a of a fairly common sort) in the example described later.

11.5.2 Selection

We first select two parent models from the current population. The parent models will be allowed to pass on some of their characteristics to a child model in the next generation. We try to select the parents in a way that favors better models (in the sense of fitness) over poorer models, but which still affords all models a reasonable opportunity to reproduce; we will return to this point later. In our algorithm, the selected parents remain in the population (that is, reproduction is not fatal).

A commonly used form of selection is illustrated in Figure (11.3). We assign a slice of a roulette wheel to each model. The size of each model's slice is somehow related to

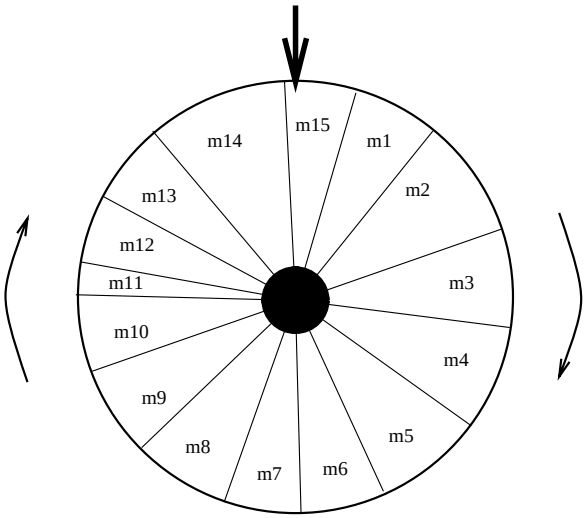

Figure 11.3: Each model in the current population gets a portion of a roulette wheel according to its rank in a population sorted by fitness. We select parents by spinning the wheel once for each parent (but we require the two parents to be distinct). (The proportions shown here are not intended to directly reflect the situation in Figure (11.2).

that model's relative fitness. In the example we discuss later "wheel space" is allocated to each member of the population as a function solely of that member's position in the population's rank order; that is, the position in the population once it has been sorted in order of fitness. We use this scheme because it is independent of many decisions about objective function scaling, but many other choices are in use.

11.5.3 Recombination

The next step is the creation of a "child" model from the selected parents. This step is in some sense the inner magic of GA's because it is mainly here that searching extends into new regions of model space. Recombination, often called *crossover*, constructs a child model by splicing together pieces copied from two parent models. The notion of "splicing" bits of models together makes sense because we explicitly regard a model as composed of a list of values. This list is called a "chromosome."

The details of this representation are extremely important to producing an effective algorithm, and the selection of an efficient representation is a major part of the art of GA's. In the example shown in Figure 11.2, where the "real" model was a single, real-valued number, a good representation might be to use a 5-bit binary number. In this case our chromosome would consist of 5 values, each of which was 0 or 1.

The simplest type of recombination, called *one-point crossover*, is as a cartoon in Figure (11.4). Here two parent models, namely 10000 and 11111, are recombined to form two

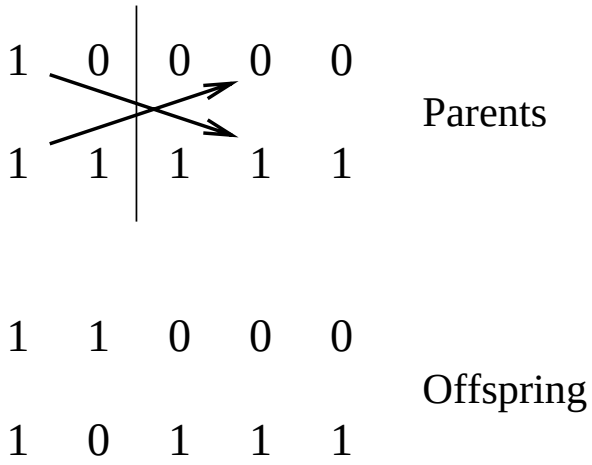

Figure 11.4: Each model is encoded as a binary string of length 5. The two upper models are the parents (and are already present in the population) and the two lower models are the offspring produced by recombination. After a crossover point is chosen randomly from the chromosome's interior, each child is constructed by taking the first portion of its chromsome from one parent and the second portion from the other.

child models, 11000 and 10111. The recombination process shown here simply requires us to pick a random intermediate point in the chromosomes, and contruct each child by selecting the first portion of its chromsome from one parent and the second portion from the other parent.

In our algorithm, one of the offspring is returned to the population to compete. At that moment, since the parents are still present in the population, there is an extra member in the population. We then discard the least fit of these. In this scheme a child must be at least as fit as the least fit pre-existing member in order to survive.

The 5-bit chromosome of binary digits can represent $2^5 = 32$ different states. There is no instrinsic reason why these had to be equally-spaced, or even ordered, points across the model domain. We are free to use mappings that are more natural to a particular problem, such as non-equally spaced points or different binary representation schemes such as (as is often done) Gray coding (see [How75]). Further, individual chromosome values do not have to be binary; many applications use larger domains (as does the one we discuss later). The selection of useful representations is one of the most challenging and central issues in making GA's useful optimizers.

Note also that recombination results in a search in model space in which newly sampled regions need not be adjacent to previously sampled regions. (Simulated annealing, on the other hand, proceeds by a series of neighbor-to-neighbor steps.) This non-linear, non-proximate search process results in a surprisingly efficient reconnaisance of model space.

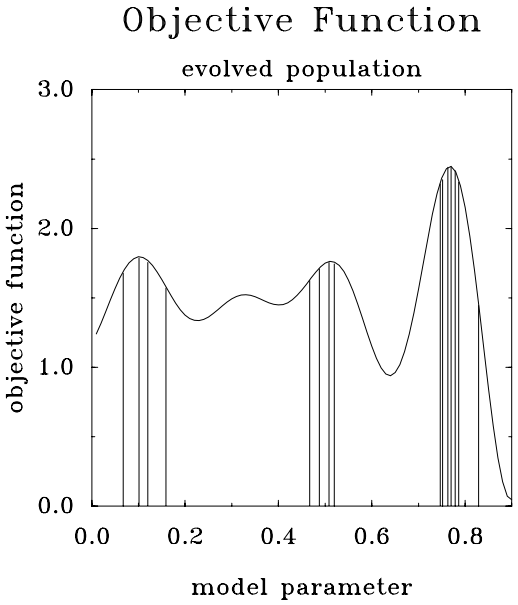

Figure 11.5: A stage in the hypothetical evolution of a GA maximization calculation associated with the model shown in Figure (11.2). The elements of the population are clustering near peaks of the objective function.

11.5.4 Mutation

Finally, most GA's also incorporate a low probability randomizing process called mutation. Mutation acts to randomly perturb a randomly chosen element in an occasional (randomly selected) child. In the absence of mutation, no child could ever acquire a chromosome gene value which was not already present in the population.

11.5.5 Search

The result of these three operations, selection, crossover, and mutation, is a new population of models the same size we started with. Figure (11.5) represents an intermediate stage in the hypothetical evolution of a GA for the optimization problem illustrated in Figure (11.2). We can see that the members of the population have begun to cluster around the maxima.

The selection process is crucial to a GA's effectiveness because it determines the balance between exploration and exploitation. A selection algorithm which gives too little weight to fitness will not lead to convergence near maxima; with no weight at all the process simply becomes a kind of unbiased random search. The other extreme, an algorithm that overemphasizes fitness, tends to converge too quickly around the one or two fittest initial models; this extreme does not search long enough to get the lay of the land.

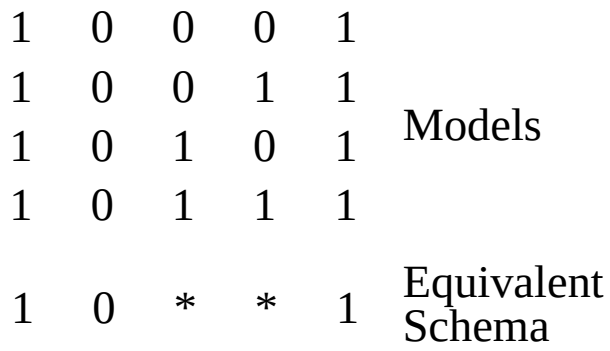

Figure 11.6: A schema represents an equivalence class of models. The asterick is a wild card or "don't care" symbol. It can take on any value in the alphabet, in this case 0 or 1. The four models shown are all associated with the schema 10**1.

There are many other issues involved in selecting a GA. Do we allow duplicate models in the population? How do we choose an initial population? How do we know when to stop? We do not have the space, the experience, or the insight to discuss any of these comprehensively. It is our experience, however, that these design issues are still open questions, and anyone who would apply these algorithms must be prepared to spend some effort in experimenting with the algorithm.

11.5.6 Schemata

Holland [Hol75] has provided a deep theoretical result that sheds light on the nature of a GA's search. Holland's insight derives from considering the effects of selection, crossover, and mutation on the probability of occurrence of *schemata* in the population.

A schema is a *regular expression* in model space. A schema is constructed by replacing zero or more of the parameters in a model with "don't care" symbols, *cf.* Figure (11.6). A schema always matches one or more actual models.

Schemata play the central role in Holland's analysis of the inner workings of GA's because schemata, unlike the models themselves, have significant and calculable chances of surviving reproduction even when the child model is different from both of its parents. Holland analyzed a simple (but representative) form of GA in terms of the change in the number of instances of a particular schema in the population. Define the fitness of a schema to be the average fitness of all of the models that are represented by that schema. Holland showed that the expected number of instances of a particular schema will grow more or less exponentially with an exponent that reflects the ratio of the fitness of that schema to the average fitness of all schemata.

Holland's insight suggests that GA's are searching for globally distributed information about the behavior of the function we seek to minimize, but that the form this information

takes is subtle. It also points out that the "state" of a GA is carried by the entire population of models, as opposed to simulated annealing in which the state is carried by one point. Thus in some sense, a GA has greater potential because it has a larger and more complex form of "memory" than simulated annealing.

11.6 Example: Array Optimization

Barth and Wunsch [BW90] proposed using optimization techniques to aid in designing acoustic tomography experiments. They applied simulated annealing to a simple yet interesting and illustrative design example. We'll apply a GA to a reduced version of their calculation.

Suppose that we wish to design a travel-time tomography experiment that is, in a sense that we will describe shortly, *maximally efficient*. To be more specific, suppose that the model we wish to determine consists of a 3×3 array of homogeneous blocks and we wish to determine the acoustic slowness of each block. Suppose further that we are given a set of potential observations; that is, a set of *potential* ray paths from which we choose the (smaller) set of actual ray paths along which we may observe travel times. Finally, suppose that we can ignore ray-bending so that all rays are straight lines. We will be allowed to make 9 actual observations, the smallest number that could possibly resolve all 9 model parameters. Our problem is to select the 9 actual rays from the set of potential rays such that the wave speeds in the model's blocks are determined as accurately as possible.

We follow Barth and Wunsch in defining "as accurately as possible" in terms of the singular values of the system's sensitivity matrix. Suppose that we have chosen a set of actual rays. Then we can compute a 9×9 matrix, S, which tells us how to compute the travel times for this set of rays, τ, given the values of the model's slowness parameters, σ:

$$\tau = S \cdot \sigma.$$

The singular values of S are a measure of the dependence of errors in the model's parameters, σ, upon errors in the travel-time data, τ. The larger the singular values of S are, the smaller the errors in σ. Our measure of goodness will be the value of the *smallest* singular value of S. (Since S is a function of the actual rays being observed, then obviously the singular values are also a function of the actual rays being observed.)

We specify the calculation by specifying the size of the model (3×3 in this case) and a set of potential rays from which the final rays must be selected. We chose a set of 216 potential rays; these are shown in Figure 11.7. (The number of geometrically feasible rays is larger than this; we reject rays that, for example, have both endpoints on the same side, or give rise to the same matrix coefficients.) Each *trial* solution of our optimization problem is a set of 9 actual rays chosen from the set of 216 potential rays. For those

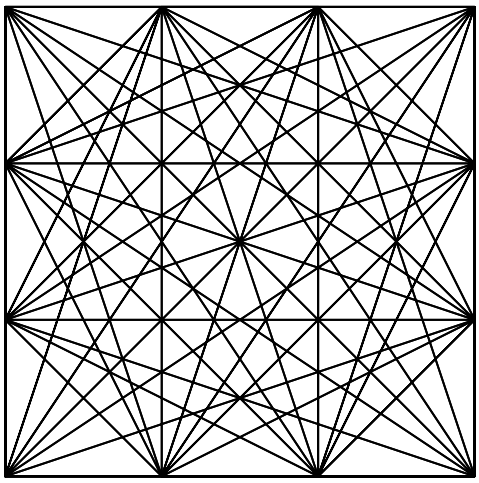

Figure 11.7: The set of 216 rays made available to a fictitious tomographic experiment. The model is a 3×3 block model. Designing the experiment requires the selection of 9 rays from this set such that the most poorly-resolved model parameters is as well-resolved as possible. Not all rays are separately resolved in the figure.

9 rays we compute a sensitivity matrix, S, and its singular value decomposition. The "goodness" of the chosen set of 9 rays is the value of the smallest singular value of S.

The function we are trying to optimize is very complicated and probably impervious to assaults by local optimization techniques. Think of each trial solution as being a point in a 9-dimensional solution space; each coordinate is integer-valued and can take on any value in the range $1, \ldots, 216$ (representing the selection of one ray from the set of potential rays). If we ponder a bit on whether or not the quantity we are trying to optimize is unimodal (has only one extremum) or not, we will be forced to the conclusion that in fact this function has *no natural shape*. Because we are free to map the integer values $1, \ldots, 216$ to the set of basis rays in any fashion we want, and because there is no "natural" mapping, we can make the shape of the function into virtually anything we wish. More to the point, we have no idea how to choose a mapping which will make this calculation amenable to local optimization.

We fed this problem to a fairly straightforward GA and allowed it to chug along for 3.5×10^5 trials. It came up with the ray set shown in Figure 11.8. We don't know if there is a better solution but this seems to be a pretty good one. Its singular values are shown in Figure 11.9.

The GA found this result after examining 3.5×10^5 ray sets. This sounds like a lot of work (and it is) but the model space it was searching contained 8×10^{20} models, so from that point of view this calculation was astoundingly efficient.

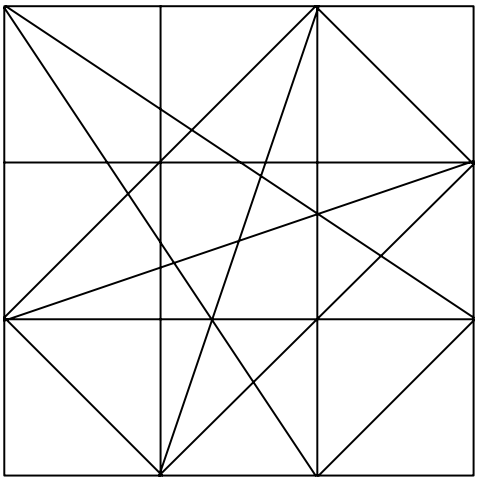

Figure 11.8: Model calculated by the GA which maximizes the smallest eigenvalue of the forward operator.

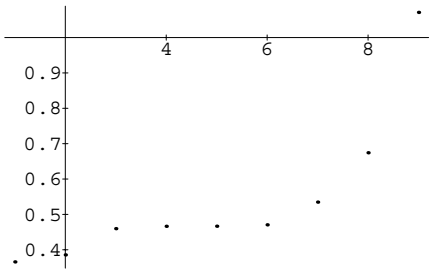

Figure 11.9: Singular value spectrum for the ray set shown in the previous figure. There are 9 singular values and the smallest is 0.37.

11.7 Summary

We don't want to leave the impression that GA's are a magic combination of a Black Box and a Silver Bullet; they're not. Casting a particular problem into a GA requires making crucial decisions about how to represent the problem domain and how to implement the basic reproductive operations. In our example here, we had to decide how to deal with the permutational degeneracy of our goodness function: all of the permutations of a particular set of actual rays have the same goodness. We also had to consider whether or not the GA crossover operation would be allowed to give rise to a child structure in which some specific ray appeared more than once: the smallest singular value for any set of actual rays with duplications is exactly zero. The point here is not how we actually dealt with these issues, but rather that the issues themselves are extremely important to the efficiency of GA's , and can easily make the difference between success and failure.

On the other hand we believe that global optimization techniques in general, and GA's in particular, will see increasing use. This popularity will reflect the increasing power of computing hardware (which makes large numerical efforts ever more practical), the increasing complexity of geophysical inverse calculations (particularly in the application of *a priori* information), and the fascinating possibilities raised by the existence of global optimization algorithms such as these.

Chapter 12

Monte Carlo, Statistical Mechanics and Combinatorial Optimization

12.1 Statistical Mechanical Foundations

As we shall see there is a deep connection between optimization by simulated annealing and the equilibrium statistical mechanics of systems of particles in a heat bath. In order to fully explain this connection it is necessary to begin with a brief excursion into classical statistical mechanics.

In Newtonian mechanics, the state of a system of point particles can be completely specified by giving the position and momentum of each particle. There are three independent coordinates or *degrees of freedom* for each of the N particles, for a total of $3N$ degrees of freedom. Thus the state of such a system can be completely specified as a point in a $6N$ dimensional Euclidean space Γ (the phase space) whose axes are the coordinates and momenta. The dynamical evolution of the system, governed by Hamilton's equations, then amounts to a path in phase space—a nonintersecting path since if the potential energy is reasonably behaved, the solutions of Hamilton's equations are unique except for isolated equilibrium configurations. The level surfaces of the Hamiltonian ($H = E$, where E is a constant) are $6N-1$ dimensional hypersurfaces ∂E of constant energy E in Γ-space. For example, the energy of a single mass-oscillator system in 1-d is just $\frac{1}{2}mv^2 + \frac{1}{2}kx^2$, so in this simple case the energy surface ∂E is just the boundary of an ellipse in the plane. If the energy of a system is constant, then all of its trajectories will lie on the appropriate surface ∂E.

For conservative mechanical systems the Hamiltonian is equal to the total energy, and $dH/dt \equiv 0$ along paths in phase space. In addition to the energy, there will be other constants of the motion. One could, for example, imagine simply integrating Hamilton's equations to achieve $6N-1$ additional constants of the motion. These constants of the

motion (or integrals) are said to be "isolating" or "not isolating" according to whether they define whole surfaces in Γ-space or not. Isolating integrals, in a sense, trap the trajectories, preventing them from moving freely about the energy surface. A system is said to be *ergodic* (or metrically transitive) if there are no isolating integrals other than the energy [Rei80]. In the example of the harmonic oscillator, it is clear that every trajectory, no matter where it starts, must travel completely around the ellipse. Thus the energy is the only isolating integral for the single oscillator system.

An idea central to almost all of statistical mechanics is that for a system in thermal equilibrium, all states with the same energy are equally probable. Thus the probability that the state of a system will lie in a given neighborhood B on the energy surface is simply the ratio of the area of the neighborhood to the area of the entire energy surface. In symbols this is expressed as

$$P(B) = \frac{\Sigma(B)}{\Sigma(E)}. \tag{12.1.1}$$

(As we shall see, this is essentially the canonical ensemble of Gibbs.) As a result, if we plot the trajectory for any initial configuration (except for a set of measure zero) we will eventually see this path come arbitrarily close to any point on the energy surface. In other words, if we cover the energy surface with disjoint open sets of arbitrarily small size, and wait long enough, the trajectory for almost any initial configuration of the system will eventually intersect all of the sets. Thus we can see intuitively that ergodicity and thermal equilibrium are intimately connected. The connection is made formally by Birkhoff's [Bir31] ergodic theorem[1] which says that a system is ergodic if for any integrable function on Γ-space, the spatial or phase average of the function over ∂E equals its temporal average.[2] Reichl [Rei80] gives examples of the application of this theorem to some simple systems.

For any macroscopic system, the number of particles N is unimaginably large; yet the measurable, thermodynamic state of our system can be described by a few variables such as temperature, pressure, flow velocities, volume, and so on. Thus from the thermodynamic viewpoint we know virtually nothing about the microscopic details of the system. The way out of this difficulty was provided by Gibbs, who had the insight that rather than concentrating on the time evolution of a single path in phase space, which would require specifying the initial positions and momenta of roughly 10^{23} molecules (per mole), one ought to look instead at an ensemble of identical systems (identical with respect to the imposed macroscopic conditions) each of whose states moves in accord with Hamilton's equations. In other words, since our knowledge of the state of the system at any instant is almost nonexistant, we must use a probability density to specify it. A probability density for a single system can be represented by an ensemble of systems, each given by a point

[1] The original ergodic hypothesis of Boltzman ([Bol71b] and [Bol71a]), that the trajectory passed through every point of ∂E, was shown to be untenable by Rosenthal [Ros13] and Plancherel [Pla13].

[2] *I.e.*, $\int_\Gamma A(q,p)\rho\, dp\, dq = \lim_{T\to\infty} \frac{1}{T}\int_0^T A(q(t),p(t))\, dt$, assuming ρ is normalized.

in Γ. Assuming the points in Γ are very dense, we can use a continuous density ρ, where

$$\rho(q_1, ..., q_f, p_1, ..., p_f, t) dq_1 \cdots dq_f, dp_1 \cdots dp_f \qquad f = 1, 2, ..., 3N \tag{12.1.2}$$

is the number of systems contained in the Γ volume element

$$(q_1, q_1 + dq_1) \cdots (p_f, p_f + dp_f) \tag{12.1.3}$$

at time t. In time, the points in the ensemble move through phase space as a fluid flows. And since systems are by definition conserved, ρ is constrained by the the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho v = 0 \tag{12.1.4}$$

where $v \equiv (\dot{q}_1, ..., \dot{q}_f, \dot{p}_1, ..., \dot{p}_f)$. Further, from Hamilton's equations

$$\dot{q}_j = \frac{\partial H}{\partial p_j}, \qquad \dot{p}_j = -\frac{\partial H}{\partial q_j}, \tag{12.1.5}$$

it follows that the "fluid" flow is incompressible:

$$\nabla \cdot v \equiv \sum_i \frac{\partial \dot{q}_i}{\partial q_i} + \sum_i \frac{\partial \dot{p}_i}{\partial p_i} = 0. \tag{12.1.6}$$

This incompressibility gives rise to Liouville's theorem $\frac{D\rho}{Dt} = 0$, or in terms of Poisson brackets, $\frac{\partial \rho}{\partial t} = -[\rho, H]$.

The precise form of the probability density ρ depends on the choice of thermodynamic variables. The constant energy ensemble is called the "microcanonical." For this case

$$\rho(q, p) = \delta(H - E) \tag{12.1.7}$$

where δ is the Dirac distribution.[3]

The constant temperature ensemble is called the "canonical" ensemble. The canonical probability density is derived by considering a very large system in diathermal contact (*i.e.*, allowing the transport of heat but not mass) with a small one. The large system is called the "heat bath" and acts to fix the temperature of the smaller system. Since the effects of the smaller system will be concomitantly small, the probability density can be developed as an expansion using the Hamiltonian of the second system as a small parameter. It turns out that a direct expansion is too restrictive, but that a logarithmic expansion is quite widely applicable ([Bec67], Ch. 37). The result is

$$\rho(q, p) = Ce^{-H(q,p)/k_B T} \tag{12.1.8}$$

[3]The normalization for this density can be shown to be $\int \rho \, dq \, dp = \int \frac{dA}{|\nabla H|}$, where dA is an element of the surface $H = E$. ([Bec67], Ch. 32)

or in terms of an energy distribution

$$\rho(E) = Q\omega(E)e^{-E/k_BT} \tag{12.1.9}$$

where k is Boltzman's constant, C and Q are normalizations, and ω is the density of states. (*Cf.* [Bec67] for more details on the definition of ω, which is potentially subtle as it depends on whether or not particles are distinguishable, *i.e.*, whether particles can be exchanged without affecting the number of states.) There are many alternative derivations of (12.1.8) and (12.1.9). Reichl ([Rei80], Ch. 9.B.2) extremizes the Gibbs entropy via Lagrange multipliers and Weiner ([Wei83], Ch. 2.7) calculates ρ directly by its definition in terms of phase space volume in the limit as the number of particles in the heat bath goes to infinity.

The temperature, actually $\frac{1}{k_BT}$, is defined to be the logarithmic derivative with respect to energy of the larger system's density of states. In thermodynamics the energy of a system is a function of its temperature. But in statistical mechanics, energy and temperature are coupled by a basic uncertainty relation. If a system is closed, then its energy is well defined (microcanonical ensemble) and the temperature is defined, subject to fluctuation, in terms of the logarithmic derivative of the density of states. On the other hand, if the system is in equilibrium and in contact with a heat bath, T is well-defined and the energy is known only approximately according to (12.1.9). The reason T is well-defined is that since the heat bath is a large system (as large as we want) its temperature fluctuations can be neglected.

12.2 Annealing As Global Optimization

Annealing is the process whereby a solid in diathermal contact with a heat bath is brought to its melting point by raising the temperature of the heat bath; after the molecules of the melted phase are allowed to arrange themselves randomly, the material is gradually cooled, step-by-step, to its freezing point. It is important that each time the temperature of the heat bath is lowered slightly, the material is allowed to come to thermal equilibrium. Eventually the material is frozen into a stable, low energy state corresponding to some periodic arrangement of the molecules in a lattice.[4] If the cooling is done too rapidly, defects (any violation of the symmetry of the lattice such as vacancies or dislocations) are frozen in, thus weakening the material. And if the cooling is done instantaneously, a process known as quenching, the system will likely freeze into an metastable, amorphous state.

If we plot the Hamiltonian of a system $H(p, q)$ (p and q represent all of the generalized momentum and position coordinates) as in Figure 12.1, then a geometrical interpretation of annealing is that it is a way of moving the system around the energy surface until it lies

[4]However, there are systems known as spin glasses where this does not apply. The study of these strange beasts is an active area of research both for condensed matter theorists and global optimizers.

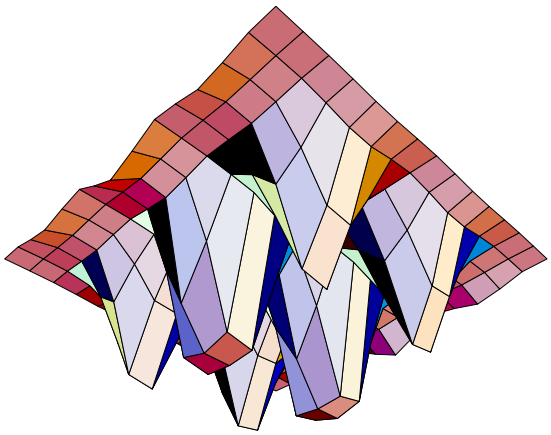

Figure 12.1: The hypothetical energy surface of a system with a two dimensional phase space. This surface has numerous local minima.

within the basin of attraction of the global minimum and then freezing it. Now, since the system can exchange energy with the heat bath, its temperature is subject to fluctuations, as previously discussed. Therefore the state of the system is not constrained to lie precisely on the surface $H = constant$ but will lie within some shell $H - \delta H \leq H \leq H + \delta H$. As the temperature goes to zero, the volume of phase space accessible to the system goes to zero as well, as all of the most probable states are concentrated near zero energy.

As we have seen, a system in a heat bath at equilibrium and at constant temperature is governed by the canonical ensemble of Gibbs ([Bec67]; [Rei80]). In this ensemble, the probability that for a given temperature T the energy will be E is given by the Boltzman distribution

$$\text{Prob}\{H = E\} = \frac{1}{Z(T)} \exp\left(\frac{-E}{k_B T}\right) \tag{12.2.1}$$

where k_B is Boltzman's constant and the normalization factor Z is called the partition function.[5] The exponential factor is such that at very low temperatures only the lowest energy states are likely. Thus annealing is one of nature's ways of finding the minimum (global or near-global) of a complex function, the Hamiltonian. This is not to say that nature always finds the global minimum, even through annealing. Many systems are known to exist in long-lived metastable states associated with local minima of the Hamiltonian. In fact the lifetime of a state associated with an energy barrier $\Delta E \gg k_B T$ is approximately

$$\tau = \tau_m \exp(\Delta E / k_B T) \tag{12.2.2}$$

where τ_m defines the macroscopic time scale (*cf.* [Sew86], Ch. 6). Because of the exponential, τ can be quite large.

[5] Invariably Z; from the German *Zustandssumme*

12.3 Markov Chains and Monte Carlo Methods

The connection between annealing and optimization is made by assuming that the objective function of our optimization problem can be interpreted as the Hamiltonian of a system in a heat bath. That being the case, simulated annealing offers a constructive method for finding the global minimum of such objective functions.

We begin by choosing an initial point ω at random in the domain of the objective function Ω. We assume for now that there are finitely many points in Ω—so-called combinatorial optimization—a restriction which is not at all severe since we can always map the domain onto a hypercube with as many nodes as are necessary to accurately represent the solution. (*I.e.*, we can assume that Ω is an undirected graph.) We then construct a random walk from this starting point $\{X_k\}$, that is, a sequence of Ω-valued random variables forming a Markov chain M with stationary, or time-independent, transition probabilities

$$P(\omega, \omega') = \text{Prob}\{X_{k+1} = \omega' | X_k = \omega\} \qquad \omega, \omega' \in \Omega. \tag{12.3.1}$$

$P(\omega, \omega')$ is a conditional probability; it is the probability that state at step $k+1$ will be ω' given that the state at step k was ω. By definition, a Markov chain is a random walk in which the higher order conditional probabilities all equal $P(\omega, \omega')$:

$$\begin{aligned} &\text{Prob}\{X_{k+1} = \omega' | X_k = \omega | X_{k-1} = w_1\} = \\ &\text{Prob}\{X_{k+1} = \omega' | X_k = \omega | X_{k-1} = w_1 | X_{k-2} = w_2\} = \\ &\vdots \\ &P(\omega, \omega'). \end{aligned} \tag{12.3.2}$$

To illustrate, consider the following simple example from Isaacson & Madsen [IM76]. Toss a fair coin repeatedly. Let X_n denote the number of heads at the n-th toss. This defines a Markov chain with a bidiagonal transition matrix having $1/2$ on the main diagonal and $1/2$ on the superdiagonal. If we want to know where the chain is at any instant, we need to know where it started: let $a_0 = (\alpha_0, \alpha_1, ..., \alpha_{n-1})$ be the initial state, subject to $\sum_i \alpha_i = 1$. If we have deterministic knowledge of the state of the system, one of the α_i will be one and the rest zero. After one time step the state of the system will be given by

$$a_1 = a_0 P. \tag{12.3.3}$$

After two time steps

$$a_2 = a_1 P = a_0 P^2. \tag{12.3.4}$$

And so on. Thus in matrix language, the evolution of this stationary random walk is governed by the Chapman-Kolmogorov identity: $P^{l+m} = P^l P^m$, where an element of the left side P_{ij}^{l+m} represents the probability of going from state i to state j in $l+m$ steps. For some stochastic matrices, the long-time probability of ending up in a particular state is independent of the starting point. This implies that the rows of P^n converge to a single

vector π such that $\pi P = \pi$. In particular, this will be true for ergodic processes. For more details see [IM76].

In our case the transition probabilities are assumed to be locally uniform and reversible. Local uniformity means that $P(\omega, \omega')$ is equal to zero if ω' is not in the neighborhood of ω and equal to one over the number of states in the neighborhood if it is. Local reversibility means that $P(\omega, \omega') = P(\omega', \omega)$, which in turn makes P a doubly stochastic matrix.[6] Now any stochastic matrix obviously has 1 as an eigenvalue. Further, for any such matrix P, the left eigenvector of unity is an equilibrium distribution (invariant probability vector: $\pi P = \pi$) of the associated Markov chain ([IM76], corollary to Theorem IV.2.1). If the Markov chain is not ergodic, then this equilibrium distribution will be associated with a subset of the chain only (an irreducible positive persistent chain). Given the definition of our Hamiltonian as the objective function of an optimization problem, it is clearly plausible to assume ergodicity; *i.e.*, that every state of the chain can be reached from every other state. In the Markov chain nomenclature, ergodicity is called irreducibility.

There can be no other nontrivial positive left eigenvectors of P since

$$\lambda x = x \cdot P \Rightarrow \lambda \sum_i x_i = \sum_i \sum_j x_j P_{ji} = \sum_j x_j \tag{12.3.5}$$

whence if $x_i \geq 0$ then $\lambda = 1$ and, conversely, if $\lambda \neq 1$ then the elements of x must sum to 0. The multiplicity of the eigenvalue 1 is equal to the number of irreducible closed subsets of the chain ([IM76], Ch. 4): once again, assuming ergodicity simplifies the discussion.

From this stationary random walk we construct a temperature dependent family of Markov chains $M(T)$, the nonstationary transition probabilities for which are given by

$$P(T;\omega,\omega') = \begin{cases} P(\omega,\omega')e^{(-\Delta E/k_B T)} & \text{when } \Delta E > 0 \\ P(\omega,\omega') & \text{when } \omega \neq \omega' \text{ and } \Delta E \leq 0 \end{cases} \tag{12.3.6}$$

and

$$P(T;\omega,\omega) = 1 - \sum_{\omega' \neq \omega} P(T;\omega,\omega'), \tag{12.3.7}$$

where $\Delta E = E(\omega') - E(\omega)$. In words, we accept unconditionally any step which decreases the energy, and we accept conditionally a step which increases the energy if a number chosen at random with uniform probability on $[0,1]$ is less than the Boltzman factor. Were it not for the ability of the method to accept unfavorable steps, the first part of (12.3.6), the random walk would immediately become trapped in a local minimum near the starting point. Equation (12.3.7) is necessary to ensure proper counting; if we do not count moves from ω to ω' which are forbidden, then we erroneously reduce the number in state ω relative to ω' [MRR+53].

[6] A stochastic matrix is a matrix whose elements are all non-negative and whose rows sum to one. The rows, therefore, may be interpreted as probability densities. A doubly stochastic matrix is one in which both the rows and columns sum to one. One can show by induction that if P is stochastic, then P^n is as well, for any n.

If the temperature T is fixed, then we are describing the Monte Carlo method of Metropolis *et al.* [MRR+53]. These authors show that the method does indeed choose configurations with a probability equal to the Boltzman probability. More precisely, they show that if local reversibility holds at infinite temperature (*i.e.*, in the absence of the Boltzman factors), then the equilibrium distribution of (12.3.6)–(12.3.7) is

$$p^0(T;\omega) \equiv \frac{1}{Z(T)} \exp\left(-\frac{E(\omega)}{k_B T}\right). \tag{12.3.8}$$

This is an example of *importance sampling*. We could achieve the same result by sampling the domain uniformly and weighting with the Boltzman factor. But most of these samples would contribute very little to the distribution. The beauty of the Metropolis algorithm is that we preferentially sample those points which are most likely to contribute.

Equations (12.3.6)–(12.3.7) are by no means the only stochastic matrix that we could choose. Allen and Tildesley [AT87] discuss other examples and the means of characterizing their relative efficiency. Although Metropolis and Ulam were the first to use the term "Monte Carlo" [MU49], random sampling of probability distributions goes back at least to the turn of the century. From Allen and Tildesley [AT87] we learn that Student (*nom de plume* of W.S. Gosset [Stu08]) computed the correlation coefficients of his distribution by random sampling. In 1901, the Italian mathematician Lazzerini experimentally verified Buffon's needle problem by dropping some 3400 needles, thereby (fortuitously, it turns out) estimating pi to be 3.1415929 (cf., [Ped58]). And Lord Kelvin [Kel01] had some 5000 random trajectories generated to study elastic collisions.

A basic result from probability theory relates the fluctuations in ensemble averages to the size of the sample. If states of the system are chosen according to (12.3.6) – (12.3.7) then ensemble averages of a given quantity A are, after n Monte Carlo steps, related to the expectation of A in the canonical ensemble by

$$\langle A \rangle_{Gibbs} = \langle A \rangle_{system} + O(n^{-1/2}). \tag{12.3.9}$$

For details, see [Chu60].

12.3.1 Historical Digression: Stanislaw Ulam, John von Neumann, and the Monte Carlo method

Stanislaw Ulam was born in 1909 in Lwów, Poland. He was educated at the Polytechnic Institute of Lwów where his mathematical teachers were the likes of Kuratowski, Mazur and Banach. According to Ulam, Banach "enjoyed long mathematical discussions with friends and students. I recall a session with Mazur and Banach at the Scottish Café that lasted seventeen hours without interuption except for meals".

Ulam began his frequent visits to the US in the late 1930's thanks in part to the good offices of John von Neumann at Princeton. He was fortunate to be on one such visit when war broke out in Europe. He remained in the US, except for brief trips, until the end of his life.

Although the first published account of the Monte Carlo method is the paper by Metropolis and Ulam [MU49], Ulam and von Neumann had exploited the ideas for several years in their work at Los Alamos. Here is Ulam's own account of the development of Monte Carlo methods from some remarks he made in 1983 [Coo].

> The first thoughts and attempts I made to practice [the Monte Carlo method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers and I immediately thought of problems on neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent from interpretable as a succession of random operations. Later ... [in 1946, I] described the idea to John von Neumann and we began to plan actual calculations.

By early 1947, von Neumann had already begun to carry out Monte Carlo calculations of neutron diffusion. And by 1948 Ulam reported to the Atomic Energy Commission about the application to such things as comic rays and the solution of Hamilton-Jacobi differential equations.

About his good friend, whom he called "Johnny", Ulam had this to say:

> His immense work stands at the crossroads of the development of exact sciences....His ideas also advanced immeasurably the attempts to formalize the new, strange world of physics in the philosophically strange work of quantum theory. Fundamental ideas of how to start and proceed with the formal modes of operations and the scope of computing machines owe an immense debt to his work, though they still today give hints that are only dimly perceived about the workings of the nervous system and the human brain itself.

12.4 Simulated Annealing

The generalization of the Metropolis Monte Carlo method to simulated annealing involves beginning the calculation at a relatively high temperature (T_∞), sufficiently high that all or most unfavorable steps are accepted, and gradually lowering T until unfavorable steps are accepted with vanishingly small probability (T_0), at which point, the system is effectively frozen. In this case the Markov chain has nonstationary transition probabilities

$$P(T_k;\omega,\omega') = \text{Prob}\{X_k + 1 = \omega' | X_k = \omega\} \qquad \omega,\omega' \in \Omega \tag{12.4.1}$$

where the set $\{T_0 \leq T_k \leq T_\infty\}$ comprises the annealing schedule.

The simulated annealing algorithm thus described (the simplest, "homogeneous" form of the algorithm) is known to converge asymptotically with probability 1 to the global minimum; a proof is given by [vLA87]. This form of the algorithm is due to [KGV83] and has been extensively tested on a number of difficult combinatorial optimization problems such as the traveling salesman problem. Many generalizations exist which allow one to take advantage of special knowledge of the objective function. For example, if the objective function possesses a global minimum which is substantially deeper than the local minima, then the generalized simulated annealing algorithm of [BJS86] may be very effective. These authors let their temperature depend on the current value of the objective function. If the global minimum is zero, then it is unlikely that the random walk will escape from the global minimum once it is entered. If the global minimum is not known *a priori* to be zero, then they suggest a scheme for achieving approximately the same result by shifting by the current smallest value of the objective function.

12.5 Calculating the Annealing Schedule

The success of simulated annealing will depend to a large degree on how the annealing schedule is chosen. Lower the temperature too fast and, as in physical annealing, the system will likely freeze in a local minimum. Lower the temperature too slowly and the computational performance will be unacceptable. In inverse scattering problems, each evaluation of the objective function requires solving a forward scattering problem; in the examples that we shall address this requires computing synthetic seismograms. Therefore relatively efficient convergence is important. The most widely used annealing schedule is the logarithmic schedule

$$T(t) = \frac{T_\infty}{\ln(1+t)}, \tag{12.5.1}$$

where t is the time-step along the Markov chain. [GG84] show that a sufficient condition for asymptotic convergence of the annealing is that the schedule not be faster than this.

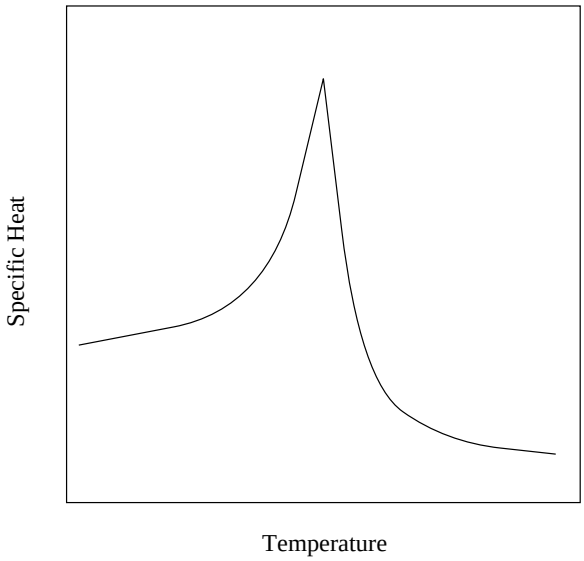

Figure 12.2: Hypothetical specific heat curve showing the critical temperature.

A much faster schedule has recently been proposed by Szu and Hartley [SH87]:

$$T(t) = \frac{T_\infty}{1+t}. \tag{12.5.2}$$

We have implemented both of these schedules and will compare their results; but our hope is that by appealing to the detailed physics of phase transitions and the theory of Markov chains, we can calculate efficient annealing schedules automatically. Recent work in [NS88] and [Mat89] points the way in this direction.

The critical temperature in real annealing is the melting point of the system. At this point a phase transition occurs as the system either gains (freezing) or loses (melting) long-range order. Thus one may be able to determine the presence of freezing or melting by looking, for example, at the critical behavior of the specific heat. Figure 12.2 shows a hypothetical example of a phase transition. For a real example, see [EGST89].

The specific heat, as well as other thermodynamic quantities such as the entropy and the Helmholtz potential, can be calculated from the partition function and the first and second moments of the Hamiltonian, increasingly accurate approximations of which are accumulated at each step of the procedure. The specific heat, for example, is just

$$C(T) = \frac{\partial \bar{E}(T)}{\partial T} \tag{12.5.3}$$

where $\bar{E}$ is the expectation value of the objective function. It can be shown that $C(T) = \sigma^2/T^2$ where σ is the variance of the energy at equilibrium. The Helmholtz free energy (or Helmholtz potential) is

$$F(T) = \bar{E}(T) - T\,Z(T) \tag{12.5.4}$$

This potential is a natural choice for studying equilibria in the canonical ensemble because of the minimum principle: The equilibrium value of any unconstrained internal parameter in a system in diathermal contact with a heat bath minimizes the Helmholtz free energy at the constant temperature of the heat bath [Cal60].

What we need to quantify are:

- How far from equilibrium is the system's distribution.
- How rapidly the temperature can be changed between steps.

On the one hand, we know that the specific heat relates the changes in internal energy to changes in temperature. On the other, we know from Markov chain theory that the rate of change of the energy with time is governed by the eigenvalues of the stochastic matrix P. Relating these two rates gives an implicit expression for the annealing schedule.

Nulton and Salamon [NS88] provide a rigorous mathematical discussion for this line of argument, speculated upon heuristically by Kirkpatrick *et al.* [KGV83], by analyzing approximately the discrepancies between the calculated distribution of X_k

$$p_k(\omega) = \text{Prob}\{X_k = \omega\} \tag{12.5.5}$$

and the equilibrium distribution (12.3.8). This discrepancy is a measure of how far from equilibrium the system is. They measure this discrepancy in terms of the function Ξ where

$$\Xi \equiv \sum_{\omega} \frac{[p(\omega) - p^0(\omega)]^2}{p^0(\omega)}. \tag{12.5.6}$$

The function Ξ generates a Riemannian metric ($ds^2 = \Xi(p, p + dp)$) on probability space "which quantifies the separation between the system $[p]$ and the target $[p^0]$ as the measure of statistical uncertainty associated with their distinguishability" [NS88]. If it is assumed that the system itself is a canonical distribution, and if the perturbations from p^0 are small then

$$\Xi = (\delta E)^2 / \sigma(E^0)^2 \tag{12.5.7}$$

where E and E^0 are the expectation values of the energy in the system and target distributions, $\delta E \equiv E - E^0$, and

$$\sigma(E^0)^2 = \sum_{\omega} \left[E(\omega) - E^0\right]^2 p^0(\omega). \tag{12.5.8}$$

Now the specific heat governs changes in energy with respect to changes in temperature, while the rate of change of energy with respect to time at constant temperature is the relaxation rate of the stationary process. To relate them [NS88] assume that

$$\frac{dE}{dt} = \frac{-1}{\epsilon} (E - E_0). \tag{12.5.9}$$

This is plausible since the relaxation of a stationary Markov Chain is asymptotically exponential. From this and (12.5.7), and using $C(T) = \sigma^2/T^2$, it follows that

$$\frac{dT}{dt} = -(vT/\epsilon)\left(v\theta + \sqrt{C(T)}\right)^{-1} \tag{12.5.10}$$

where $\theta(T) = 1 + (T/2C)\frac{dC}{dT}$ and $v \equiv \sqrt{\Xi(p^0, p)}$.

Equation (12.5.10) is implicitly the annealing schedule. It remains to be seen in practice whether the assumptions involved in (12.5.9) will be justified in practice, although it seems reasonable for slow annealing schedules. In order to utilize (12.5.10) in practice we need to be able to calculate the specific heat C and the stationary relaxation rate ϵ. Calculating C is straightforward using $C = \Sigma^2/T$, since one readily accumulates approximations to $\bar{E}$ and $\bar{E^2}$. The stationary relaxation rate ϵ is more difficult. From the Perron-Frobenius theory ([Sen81]; [IM76]) we know that ϵ is related to the penultimate eigenvalue of P. But for a realistic problem, say 100 unknowns and 100 lattice points for each unknown, P is of order $100^{100} \times 100^{100}$. On the other hand, since Ω has the topology of a hypercube we know that **only** $2 \times 100 \times 100^{100}$ of these elements will be nonzero. Relying on the principle of importance sampling, it is at least plausible to suppose that a reasonable approximation to ϵ can be computed at each Monte Carlo phase of the procedure. (For example, one may use Lanczos' method or conjugate gradient [Sca89].) Nulton and Salamon [NS88] provide an alternative method of estimating ϵ approximately, which we have tentatively implemented. An analysis of the validity of this approximation will be the subject of a future report. For starters, however, there is an easy way of achieving approximately the same end within the framework of any of the conventional annealing schedules, and that is to make an accurate calculation of the specific heat function and use that to identify the critical temperature where the quasi-static annealing should begin. (We simply plot the specific heat and look for peaks.) To do this you must make an ensemble of trail annealing runs to get accurate statistics. We usually do at least 10 and take the ensemble average of the individual specific heat functions. Once this is done, any sufficiently slow annealing schedule appears likely to work.

12.6 Surfaces and Volumes in High-dimensional Spaces

The discussion of optimization/sampling of functions of many dimensions brings up a result from statistical mechanics that never ceases to fascinate, as it is highly counter-intuitive. It concerns spaces of very high dimensionality. These arise in statistical mechanics where it is theoretically convenient to consider an entire system of N particles as representing a point in something called phase space. For each particle there are 3 spatial coordinates and 3 momemta, for a total of $6N$ coordinates. You can well imagine that

with Avogadro's number of molecules in a reasonable-sized sample of a gas, that phase spaces get pretty big.

OK, so no matter what the dimension, we can say that the volume of a sphere of radius r in ν dimensions is

$$V(r) = C_\nu r^\nu$$

where C_ν depends only on the dimension.

So the volume of a spherical shell V_s of thickness s is given by

$$V_s = V(r) - V(r-s) = C_\nu(r^\nu - (r-s)^\nu) = V(r)\left(1 - \left(1 - \frac{s}{r}\right)^\nu\right).$$

Now for $s/r < 1$ and $\nu \gg 1$ we have

$$V_s \approx V(r)\left(1 - e^{-\nu s/r}\right).$$

So imagine what happens if ν is on the order of 10^{20} or more: **the total volume of a sphere is already contained in an unimaginably thin skin below the surface of the sphere.**

And how about that constant C_ν above? Not that it matters to the argument, but you never know when you're going to need the volume of an $n-$dimensional sphere, so here's how you do it.

Consider the ν-dimensional integral

$$J = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-x_1^2 + \cdots x_\nu^2}\, dx_1 \cdots dx_\nu.$$

As bad as this may look, it turns out to be easy to compute.

$$J = \left\{\int_{-\infty}^{\infty} e^{-x^2}\, dx\right\}^\nu = \pi^{\nu/2}$$

On the other hand, we can express J in spherical coordinates

$$J = \int_0^\infty e^{-r^2} r^{\nu-1} \Omega_\nu\, dr = \Omega_\nu \int_0^\infty e^\eta \eta^{\nu/2-1} \frac{d\eta}{2} = \frac{\Omega_\nu}{2}\Gamma(\nu/2)$$

where Ω_ν is the surface of the ν-dimensional sphere and Γ is the gamma function. (For even arguments, the gamma function is equal to the factorial.) Comparing these two results we have

$$\Omega_\nu = \frac{2J}{\Gamma(\nu/2)} = \frac{2\pi^{\nu/2}}{\Gamma(\nu/2)} = \frac{2\pi^{\nu/2}}{(\nu/2-1)!}$$

assumeing ν is even, which we may safely do.

On the other hand we have directly that

$$V_\nu(R) = \int_0^R r^{\nu-1}\Omega_\nu \; dr = R^\nu \frac{\Omega_\nu}{\nu} = \frac{\pi^{\nu/2}}{(\nu/2)!}R^\nu.$$

This result comes up frequently in conversation at the tonier cocktail parties, so don't be afraid to inject it into conversations with your friends and family.

Bibliography

[AT87] M.P. Allen and P.J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1987.

[Bar76] Robert G. Bartle. *The elements of real analysis.* Wiley, 1976.

[BB91] Richard K. Belew and Lashon B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA, 1991. Morgan Kaufmann Publishers, Inc.

[Bec67] R. Becker. *Theory of Heat.* Springer-Verlag, N.Y., 1967.

[Bir31] G.D. Birkhoff. *Proceedings National Acadamy of Science (U.S.)*, 17:656, 1931.

[Bjö75] A. Björk. Methods for sparse linear least-squares problems. In J. Bunch and D. Rose, editors, *Sparse Matrix Computations.* Academic, New York, 1975.

[BJS86] I.O. Bohachevsky, M.E. Johnson, and M.L. Stein. Generalized simulated annealing. *Technometrics*, 28:209–217, 1986.

[Bol71a] L. Boltzman. Einige allgemeine Satze über Warmegleichgewicht. *Sitzber. Akad. Wiss. Wien*, 63:679–711, 1871.

[Bol71b] L. Boltzman. Über das Warmegleichgewicht zwischen mehratomigen Gasmolekülen. *Sitzber. Akad. Wiss. Wien*, 63:397–418, 1871.

[Bra90] R. Branham. *Scientific Data Analysis.* Springer-Verlag, 1990.

[BS83] P. Bloomfield and W. Steiger. *Least absolute deviations.* Birkhäuser, Boston, 1983.

[BW90] Norman Barth and Carl Wunsch. Oceanographic experiment design by simulated annealing. *Journal of Physical Oceanography*, 20(9):1249–1263, September 1990.

[Cal60] H.B. Callen. *Thermodynamics.* Wiley, N.Y., 1960.

[Cen82] Y. Censor. Row action methods for huge and sparse systems and their applications. *SIAM Review*, 23:444–466, 1982.

[Cha78] R. Chandra. *Conjugate gradient methods for partial differential equations.* PhD thesis, Yale University, New Haven, CT, 1978.

[Chu60] K.L. Chung. *Markov Chains with Stationary State Probabilities Vol. 1.* Springer, Heidelberg, 1960.

[CM79] S. Campbell and C. Meyer. *Generalized inverses of linear transformations.* Pitman, London, 1979.

[Coo] Necia Grant Cooper, editor. *From cardinals to chaos.* Cambridge University Press.

[CW80] J. Cullum and R. Willoughby. The Lanczos phenomenon—an interpretation based upon conjugate gradient optimization. *Linear Albegra and Applications*, 29:63–90, 1980.

[CW85] J. Cullum and R. Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations.* Birkhäuser, Boston, 1985.

[Dav91] Lawrence Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, NY, 1991.

[Dui87] J. Duijndam, A. *Detailed Bayesian inversion of seismic data.* PhD thesis, Technical University of Delft, 1987.

[EGST89] K. Ema, C.W. Garland, G. Sigau, and Nguyen Huu Tinh. Heat capacity associated with nematic—smectic-A_1—smectic-$\tilde{A}$—smectic-A (crenelated)—smectic-A_2 phase sequence. *Physical Review*, A39:1369–1375, 1989.

[FHW49] L. Fox, H. Huskey, and J. Wilkinson. Notes on the solution of algebraic linear simultaneous equations. *Q. J. Mech. Appl. Math*, 1:149–173, 1949.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Proceedings Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization, & Machine Learning.* Addison-Wesley, 1989.

[GR70] G. Golub and C. Reinsch. Singular value decomposition. *Numerische Math.*, 14:403–420, 1970.

[Gre87] John J. Grefenstette, editor. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, NJ, 1987. Lawrence Erlbaum Associates.

[GvL83] G. Golub and C. van Loan. *Matrix Computations.* Johns Hopkins, Baltimore, 1983.

[Hes51] M. Hestenes. Iterative methods for solving linear equations. Technical report, National Bureau of Standards, 1951.

[Hes75] M. Hestenes. Pseudoinverses and conjugate gradients. *Communications of the ACM*, 18:40–43, 1975.

[Hes80] M. Hestenes. *Conjugate direction methods in optimization.* Springer, Berlin, 1980.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[How75] Howard W. Sams & Co., Inc. *Reference Data for Radio Engineers*, 6th edition, 1975.

[HS52] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *NBS J. Research*, 49:409–436, 1952.

[HS71] Norman B. Haaser and Joseph A. Sullivan. *Real analysis.* Van Norstrand Reinhold, 1971.

[IM76] D. L. Isaacson and R. W. Madsen. *Markov Chains: Theory and Applications.* Wiley, New York, 1976.

[Jef83] C. Jeffreys, R.˙ *The logic of decision.* University of Chicago, 1983.

[Kaz37] S. Kazcmarz. Angeläherte auflösung von system linearer gleichungen. *Bull. Acad. Polon. Sci. Lett.*, A:355–357, 1937.

[Kel01] Lord Kelvin. Nineteenth century clouds over the dynamical theory of heat and light. *Phil. Mag.*, 2:1–40, 1901.

[Ker78] D. Kershaw. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26:43–65, 1978.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[Läu59] P. Läuchli. Iterative Lösung und Fehlerabschätzung in der Ausgleichsrechnung. *Zeit. angew. Math. Physik*, 10:245–280, 1959.

[Law73] C. Lawson. Sparse matrix methods based on orthogonality and conjugacy. Technical Report 33-627, Jet Propulsion Laboratory, 1973.

[Man80] T. A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Mathematics of Computation*, 34:473–497, 1980.

[Mat89] I. Matsuba. Optimal simulated-annealing method based on stochastic-dynamic programming. *Physical Review*, A39:2635–2642, 1989.

[MF53] Philip M. Morse and Herman Feshbach. *Methods of theoretical physics.* McGraw Hill, 1953.

[MRR+53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[MU49] N. Metropolis and S. Ulam. The Monte Carlo method. *J. Am. stat. Ass.*, 44:335–41, 1949.

[NS88] J.D. Nulton and P. Salamon. Statistical mechanics of combinatorial optimization. *Physical Review*, A37:1351–1356, 1988.

[Pai71] C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, London, England, 1971.

[Par60] E. Parzen. *Modern probability theory and its applications.* Wiley, 1960.

[Par80] B. Parlett. *The symmetric eigenvalue problem.* Prentice-Hall, 1980.

[Ped58] D.S. Pedoe. *The Gentle art of Mathematics.* Penguin, N.Y., 1958.

[Pis84] S. Pissanetsky. *Sparse matrix technology.* Academic, N.Y., 1984.

[Pla13] M. Plancherel. Beweis der Unmöglichkeit ergodischer mechanischer Systeme. *Ann. Phys.*, 42:1061–1063, 1913.

[PS82] C. Paige and M Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8:43–71, 1982.

[Rei80] L. Reichl. *A Modern Course in Statistical Physics.* University of Texas Press, Austin, 1980.

[Ros13] A. Rosenthal. Beweis der Unmöglichkeit ergodischer Gassysteme. *Ann. Phys.*, 42:796–806, 1913.

[SB80] J. Stoer and R. Bulirsch. *Introduction to numerical analysis.* Springer, N.Y., 1980.

[Sca89] J.A. Scales. Using conjugate gradient to calculate the eigenvalues and singular values of large, sparse matrices. *Geophysical Journal*, 97:179–183, 1989.

[Sch89] J. David Schaffer, editor. *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, CA, 1989. Morgan Kaufmann Publishers, Inc.

[SDG90] J.A. Scales, P. Docherty, and A. Gersztenkorn. Regularization of nonlinear inverse problems: imaging the near-surface weathering layer. *Inverse Problems*, 6:115–131, 1990.

[Sen81] E. Seneta. *Non-negative Matrices and Markov Chains.* Springer, New York, 1981.

[Sew86] G.L. Sewell. *Quantum Theory of Collective Phenomena.* Oxford University Press, New York, 1986.

[SG88] J. A. Scales and A. Gersztenkorn. Robust methods in inverse theory. *Inverse Problems*, 4:1071–1091, 1988.

[SGT88] J. A. Scales, A. Gersztenkorn, and S. Treitel. Fast ℓ_p solution of large, sparse linear systems: application to seismic travel time tomography. *Journal of Computational Physics*, 75:314–333, 1988.

[SH87] H. Szu and R. Hartley. Fast simulated annealing. *Physics Letters A*, 122:157–162, 1987.

[Sti52] E. Stiefel. Über einige Methoden der Relaxationsrechnung. *Zeit. angew. Math. Physik*, 3, 1952.

[Str88] Gilbert Strang. *Linear algebra and its application.* Saunders College Publishing, Fort Worth, 1988.

[Stu08] Student. The probable error of a correlation coefficient. *Biometrika*, 6:302–310, 1908.

[Tar87] Albert Tarantola. *Inverse Problem Theory.* Elsevier, New York, 1987.

[vLA87] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Practice.* Reidel, Dordrecht, 1987.

[Wei83] J.H. Weiner. *Statistical Mechanics of Elasticity.* Wiley-Interscience, New York, 1983.

[You71] D. M. Young. *Iterative solution of large linear systems.* Academic, N.Y., 1971.