Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. James C. Spall

ISBN: 0-471-33052-3

CHAPTER 8

ANNEALING-TYPE ALGORITHMS

This chapter is devoted to a class of algorithms aimed at the global optimization problem. The common theme in these algorithms is the principle of *annealing*, where the magnitudes of random perturbations are reduced—annealed—in a controlled manner. These random perturbations are injected in a Monte Carlo fashion. The annealing is designed to enhance the likelihood of avoiding local minima en route to a global minimum of $L(\boldsymbol{\theta})$. The injected randomness helps prevent premature convergence to a local minimum by providing a greater "jumpiness" to the algorithm. The term *annealing* comes from analogies to the controlled cooling of physical substances to achieve a type of optimal state for the substance.

This chapter considers two classes of annealing algorithms. Sections 8.1–8.3 discuss the popular simulated annealing algorithm, which is derived from a probability expression (the Boltzmann–Gibbs distribution) governing the energy state of a system at a fixed temperature. The second class of annealing algorithms, considered in Section 8.4, is based on the principles of stochastic approximation. Section 8.5 contains a summary and conclusions and Section 8.6 is an appendix that provides some theoretical justification for the simulated annealing algorithm via connections to stochastic approximation.

8.1 INTRODUCTION TO SIMULATED ANNEALING AND MOTIVATION FROM THE PHYSICS OF COOLING

The simulated annealing (SAN) algorithm continues in the spirit of the stochastic approximation (SA) algorithms of Chapters 6 and 7 in working with only loss function measurements (versus requiring gradient information). Even more generally, it applies in *both* problems of continuous $\boldsymbol{\theta}$ and discrete $\boldsymbol{\theta}$, as introduced in Chapter 1 (although the gradient was not used in the methods of Chapters 6 and 7, it was still assumed to exist). SAN was originally developed for discrete optimization problems, but more recently has found application in continuous optimization problems of the type emphasized in most of the previous chapters. The algorithm is designed to traverse local minima en route to a global minimum of $L = L(\boldsymbol{\theta})$.

The term *annealing* comes from analogies to the cooling of a liquid or solid. A central issue in statistical mechanics is analyzing the behavior of

substances as they cool. At high temperatures, molecules have much mobility, but as the temperature decreases, this mobility is lost and the molecules *may* tend to align themselves in a crystalline structure. This aligned structure is the minimum energy state for the system. Note the qualifier "may": Temperature alone does not govern whether the substance has reached a minimum energy state. To achieve this state, the cooling must occur at a sufficiently slow rate. If the substance is cooled at too rapid a rate, an amorphous (or "polycrystalline") state may be reached that is not a minimum energy state of the substance. The principle behind annealing in physical systems is the *slow* cooling of substances to reach the minimum energy state.

In optimization, the analogy to a minimum energy state for a system is a minimizing value of the loss function. The technique of SAN attempts to mathematically capture the process of controlled cooling associated with physical processes, the aim being to reach the lowest value of the loss function in the face of possible local minima. As with the physical cooling process, whereby temporary higher-energy states may be reached as the molecules go through their alignment process, SAN also allows temporary increases in the loss function as the learning process captures the information necessary to reach the global minimum. A more thorough explanation of the analogy between SAN and physical cooling is given, for example, in Kirkpatrick et al. (1983).

It is clear that the critical component of a SAN algorithm is the mathematical analogue of the rate of cooling in physical processes. As with other stochastic search and optimization algorithms, the choice of this implementation- and problem-specific cooling schedule (analogous to the gain coefficients in stochastic approximation) has a strong effect on the success or failure of SAN.

A primary distinction between SAN and the majority of other optimization approaches (including those discussed in earlier chapters) is the willingness to give up the quick gain of a rapid decrease in the loss function by allowing the possibility of temporarily increasing the value of the loss function. SAN derives this property from the Boltzmann–Gibbs probability distribution of statistical mechanics, describing the probability of a system having a particular discrete energy state:

$$P(\text{energy state} = x) = c_T \exp\left(-\frac{x}{c_b T}\right), \tag{8.1}$$

where $c_T > 0$ is a normalizing constant, $c_b > 0$ is known as the Boltzmann constant, and T is the temperature of the system. Note that at high temperatures, the system is more likely to be in a high-energy state than at low temperatures.

The optimization analogy derives from the fact that even at a low temperature (equivalent to the optimization algorithm having run for some time with a decreasing temperature), there is some nonzero probability of reaching a higher energy state (i.e., higher level of the loss function). So, the SAN process sometimes goes uphill, but the probability of an uphill step decreases as the

temperature is lowered. Hence, there is the possibility of getting out of a local minimum in favor of finding a global minimum. This possibility is especially prominent in the early iterations when the temperature is high.

It was Metropolis et al. (1953) who first introduced the Boltzmann–Gibbs distribution-based idea into numerical analysis through constructing a means for simulation of a system at some fixed temperature. In particular, if a system is in some current energy state $\mathcal{E}_{\text{curr}}$, and some system aspects are changed to make the system potentially achieve a new energy state $\mathcal{E}_{\text{new}}$, then the Metropolis simulation always has the system go to the new state if $\mathcal{E}_{\text{new}} < \mathcal{E}_{\text{curr}}$. On the other hand, if $\mathcal{E}_{\text{new}} \geq \mathcal{E}_{\text{curr}}$, then the probability of the system going to the new state is

$$\exp\left(-\frac{\mathcal{E}_{\text{new}} - \mathcal{E}_{\text{curr}}}{c_b T}\right). \tag{8.2}$$

Expression (8.2) is known as the *Metropolis criterion*. After a large number of such decisions and outcomes, the system eventually reaches an equilibrium where the system state is governed by the Boltzmann–Gibbs distribution in (8.1). This is predicated on the system being at the fixed temperature T.

Let us outline the arguments showing how repeated application of (8.2) leads to (8.1). This result is central to SAN and to the Metropolis–Hastings version of Monte Carlo sampling in Chapter 16. Suppose that $\mathcal{E}_{\text{new}}$ and $\mathcal{E}_{\text{curr}}$ can each take on one of N values $x_1, x_2, \ldots, x_N$. The aim, therefore, is to show that many applications of (8.2) yield $P(\text{energy state} = x_i) = c_T \exp[-x_i/(c_b T)]$ for each $i = 1, 2, \ldots, N$. The probability of going from the current state i to any other state j is the product of two probabilities: (i) the probability of *choosing* j as the next state given that the process is in state i, and (ii) given that state j is chosen, the probability that the process *actually goes* to state j from i. A common simplification—adopted here—is that the probability in (i) of choosing any one state from any other state is equal ($= 1/N$). Expression (8.2) provides the acceptance probability in (ii).

Given the above, the aim is to find the state transition probabilities, say p_{ij}, of going from state i to state j. These transition probabilities are the components of a Markov transition matrix (see Appendix E). Let a_{ij} denote the probability of accepting the transition from state x_i to state x_j. In particular, the acceptance probability for $i \neq j$ from (8.2) is

$$a_{ij} = \begin{cases} 1, & x_j < x_i, \\ \exp\left(-\dfrac{x_j - x_i}{c_b T}\right), & x_j \geq x_i. \end{cases} \tag{8.3}$$

The matrix $\boldsymbol{P} = [p_{ij}]$ represents the transition matrix. For $i \neq j$, $p_{ij} = a_{ij}/N$. Note that there is a positive probability that the process stays in state i even when the energy is lower in state j. Eqn. (8.3) does not apply to the diagonal elements p_{ii}. However, the transition probabilities must satisfy the following balance equation since each row in the transition matrix must sum to unity (because there must be a transition from state i to *some* state):

$$\begin{aligned} p_{ii} &= 1 - \sum_{j \neq i} p_{ij} \\ &= 1 - \frac{1}{N} \sum_{j \neq i} a_{ij} . \end{aligned}$$

The transition matrix providing the probabilities of going from state i to state j is therefore

$$\boldsymbol{P} = [p_{ij}] = \begin{bmatrix} 1 - \frac{1}{N} \sum_{k \neq 1} a_{1k} & \frac{1}{N} a_{12} & \cdots & \frac{1}{N} a_{1N} \\ \frac{1}{N} a_{21} & 1 - \frac{1}{N} \sum_{k \neq 2} a_{2k} & \cdots & \frac{1}{N} a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N} a_{N1} & \frac{1}{N} a_{N2} & \cdots & 1 - \frac{1}{N} \sum_{k \neq N} a_{Nk} \end{bmatrix} . \tag{8.4}$$

The transition matrix in (8.4) can be used to find the stationary distribution representing the probabilities of the system energy being in any of the N states. Let this stationary distribution be the N-dimensional vector $\overline{\boldsymbol{p}}$, where the ith element corresponds to the probability of the energy being equal to x_i. As in Appendix E, this vector satisfies $\overline{\boldsymbol{p}}^T = \overline{\boldsymbol{p}}^T \boldsymbol{P}$. This expression yields N equations and N unknowns (the elements $\overline{p}_i$). If it is assumed that the total energy in the system is fixed, that is, $\sum_{i=1}^{N} x_i$ = constant, then the solution to each of these N equations corresponds to the Boltzmann–Gibbs distribution (8.1) (see Exercise 8.1). That is, $\overline{p}_i = P(\text{energy state} = x_i) = c_T \exp[-x_i/(c_b T)]$ for all i, as we set out to show.

8.2 SIMULATED ANNEALING ALGORITHM

8.2.1 Basic Algorithm

Using the principles discussed in Section 8.1, we now present the general form of the SAN algorithm. Kirkpatrick et al. (1983) introduced the modern SAN algorithm through use of the Metropolis criterion in (8.2) for the purpose of

optimization. One of the key innovations is to use (8.2) together with the idea of a *changing* temperature that decays according to an annealing schedule (exponentially in the case of Kirkpatrick et al., 1983).

There is no single SAN algorithm; rather there are variations depending on the implementation of the annealing schedule and choice of sampling required for generating a new candidate point. As before, we consider the minimization of some loss function $L(\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$. Below are the general steps in SAN when there are perfect (noise-free) measurements of L.

SAN Algorithm with Noise-Free Loss Measurements

Step 0 (**Initialization**) Set an initial temperature T and initial parameter vector $\hat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_{\text{curr}} \in \Theta$; determine $L(\boldsymbol{\theta}_{\text{curr}})$.

Step 1 Relative to the current value $\boldsymbol{\theta}_{\text{curr}}$, randomly determine a new value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{\text{new}} \in \Theta$, and determine $L(\boldsymbol{\theta}_{\text{new}})$.

Step 2 Compare the two L values above via the Metropolis criterion (8.2). Let $\delta = L(\boldsymbol{\theta}_{\text{new}}) - L(\boldsymbol{\theta}_{\text{curr}})$. If $\delta < 0$, accept $\boldsymbol{\theta}_{\text{new}}$. Alternatively, if $\delta \geq 0$, accept $\boldsymbol{\theta}_{\text{new}}$ only if a uniform (0, 1) random variable U (generated by Monte Carlo) satisfies $U \leq \exp[-\delta/(c_b T)]$. (Without loss of generality, one can set $c_b = 1$ since the user has full control over T.) If $\boldsymbol{\theta}_{\text{new}}$ is accepted, then $\boldsymbol{\theta}_{\text{curr}}$ is replaced by $\boldsymbol{\theta}_{\text{new}}$; else, $\boldsymbol{\theta}_{\text{curr}}$ remains as is.

Step 3 Repeat steps 1 and 2 for some period until either the budget of function evaluations allocated for that T has been used or the system reaches some state of equilibrium.

Step 4 Lower T according to the annealing schedule and return to step 1. Continue the process until the total budget for function evaluations has been used or some indication of convergence is satisfied (analogous to the system being frozen in its minimum energy state). The final estimate is $\hat{\boldsymbol{\theta}}_n$ (taken as the most recent $\boldsymbol{\theta}_{\text{curr}}$), representing the $\boldsymbol{\theta}$ value after n iterations ($= n + 1$ loss evaluations).

The specifics of implementation for the steps above can vary greatly. In the annealing schedule, Kirkpatrick et al. (1983), Press et al. (1992, p. 452), and Brooks and Morgan (1995) discuss the case where T decays geometrically in the number of cooling phases (number of times T is lowered according to step 4). Specifically, for some $0 < \lambda < 1$, the new temperature is related to the old temperature according to $T_{\text{new}} = \lambda T_{\text{old}}$. Others recommend temperatures that decay at every iteration (analogous to the decaying gain sequences of SA in Chapters 4–7). One of the most common such forms has the temperature decaying at a rate proportional to $1/\log k$, where k (as usual) is the iteration index (Geman and Geman, 1984; Hajek, 1988). Szu and Hartley (1987) present arguments justifying a faster decay rate proportional to $1/k$. In practice, the analyst will rarely know a priori what temperature decay rate is best for the

problem at hand. So the analyst will typically pick a rate relatively arbitrarily and then tune the unknown coefficients associated with that rate (e.g., λ and the length of time a temperature is valid in the geometric case; the constant of proportionality in the continuous-decay rates such as $1/\log k$ or $1/k$).

Another area for different implementations is in step 1, where a new $\boldsymbol{\theta}$ value is generated randomly. Suppose that $\boldsymbol{\theta}_{\text{new}}$ is generated by adding a random perturbation to the current value $\boldsymbol{\theta}_{\text{curr}}$. Probably the most common form of perturbation for continuous optimization is to add a p-dimensional Gaussian random variable to $\boldsymbol{\theta}_{\text{curr}}$ (e.g., Jang et al., 1997, p. 183). Alternative forms include changing only one component of $\boldsymbol{\theta}$ at a time (Brooks and Morgan, 1995) (the component may be chosen either deterministically or randomly), adding spherically uniform distributed perturbations (Bohachevsky et al., 1986), and adding multivariate Cauchy distributed perturbations (Szu and Hartley, 1987; Styblinski and Tang, 1990; the scalar Cauchy distribution is discussed in Exercise C.3 of Appendix C).

Aside from the general variations in implementation above, SAN is critically dependent on the specific values for various algorithm parameters and decision criteria. In particular, these include the initial temperature T, the specific distribution parameters chosen for the perturbation distribution (e.g., the covariance matrix for a Gaussian perturbation), the specific parameter(s) associated with the decay of T (e.g., the λ in $T_{\text{new}} = \lambda T_{\text{old}}$ if one is using a geometric decay), and the criterion for determining when to drop the temperature (e.g., the maximum allowable number of function evaluations before a lowering of T).

There is some formal convergence theory governing the behavior of the SAN algorithm. For problems with discrete $\boldsymbol{\theta}$, Hajek (1988, Theorem 1) gives conditions such that SAN converges in probability to the set of global minima Θ^* for a temperature decaying at a rate proportional to $1/\log k$. For the Hajek result to apply, the constant of proportionality in the temperature decay is chosen in a manner related to the relative closeness of the loss values for the local and global minima. For problems where $\boldsymbol{\theta}$ is continuous, convergence theory for SAN may be built from the theory for stochastic approximation. This theory is similar to that for the SA algorithms in Chapters 4–7, except that the theory pertains to the *global* optimization problem. The chapter appendix (Section 8.6) summarizes this theory. Some useful analytical background for interpreting this theory is given in Section 8.4 on annealing algorithms based on SA algorithms with injected (Monte Carlo) randomness. The essential idea is that SAN is expressed as an SA algorithm where the effective noise in the SA process is given a special interpretation that is tied to the Metropolis accept/reject step (step 2 above). This effective SA noise is not to be confused with the usual SA noise involving corrupted loss or gradient measurements (the SAN theory is for noise-free loss measurements).

8.2.2 Modifications for Noisy Loss Function Measurements

There appears to be no single widely accepted way to accommodate noisy loss function measurements in SAN (similar to the problems in the random search techniques and the Nelder–Mead simplex algorithm of Chapter 2).[1] A limited amount of work has been published on this problem. For example, in the *discrete* θ setting, Gelfand and Mitter (1989), Fox and Heine (1995), and Alrefaei and Andradóttir (1999) establish some convergence properties with noisy function measurements. Ahmed and Alkhamis (2002) use a ranking and selection method (as in Sections 12.5 and 14.5) embedded within SAN to cope with the randomness.

The primary difficulty with noise arises in the critical decision step 2, where θ_{curr} is changed or not changed based on the value of $\delta = L(\theta_{new}) - L(\theta_{curr})$ together with the random sampling associated with the Metropolis criterion. With noisy function measurements, the value of δ equals $y(\theta_{new}) - y(\theta_{curr})$ rather than $L(\theta_{new}) - L(\theta_{curr})$, and, in general, these two differences will not be equal. In fact, even a small level of noise will frequently alter the sign of δ, which is likely to change the decision regarding the acceptance/rejection of the new point θ_{new}.

The most obvious method for coping with noise is to average several function measurements $y(\cdot)$ at each of the values θ_{curr} and θ_{new} when performing the comparison in step 2. However, this may dramatically increase the cost of optimization (specifically, the total number of function evaluations required) since a high amount of averaging is generally required to effectively remove the noise, especially in the region around a local or global minimum where the function may be relatively flat and the difference $L(\theta_{new}) - L(\theta_{curr})$ is likely to be small. Note also that this type of per-iteration averaging is contrary to the spirit of stochastic approximation, as mentioned in Section 4.1, where efficiencies result from averaging across iterations.

An alternative way to cope with noise is to alter the acceptance criterion to accept the inherent errors that will be made with the noisy measurements. Hence, we replace the criterion $\delta < 0$ or $\delta \geq 0$ with $\delta < \tau$ or $\delta \geq \tau$, where τ is some threshold value that may be positive or negative depending on the circumstances. Likewise, the Metropolis criterion is modified so that $\exp[-\delta/(c_b T)]$ in step 2 is replaced by $\exp[-(\delta - \tau)/(c_b T)]$. (It will sometimes be useful to express τ as a multiple of the measurement noise standard deviation as described in Proposition 8.1 below. This facilitates making approximate probabilistic arguments about the likelihood of a good or bad decision in step 2

[1] One of the particularly appealing features of the stochastic approximation methods in Chapters 4–7 is that they handle noisy function measurements in an essentially seamless manner. Namely, the forms of the algorithms do not have to be altered, the convergence theory applies in the noisy case, and there is a *gradual* degradation in performance as the noise level increases from zero (versus the potentially large degradation in approaches such as SAN that use explicit decision criteria based on loss function measurements).

of the algorithm.) In particular, if it is believed that there are many local minima, then τ should be taken positive. The rationale for such a change is that while we are willing to accept a $\boldsymbol{\theta}_{new}$ that temporarily increases the loss function (a basic aspect of SAN!), we are less willing to forgo a $\boldsymbol{\theta}_{new}$ that decreases the loss function. Changing the unconditional acceptance criterion from $\delta < 0$ to $\delta < \tau$ with $\tau > 0$ will allow for a greater number of cases where $L(\boldsymbol{\theta}_{new}) < L(\boldsymbol{\theta}_{curr})$ even though $y(\boldsymbol{\theta}_{new}) \geq y(\boldsymbol{\theta}_{curr})$ due to the noise. If τ is large enough (say, several times the standard deviation of the measurement noise), one can be sure that most such combinations will be caught. This will be at the inevitable expense of letting some additional $\boldsymbol{\theta}_{new}$ values that increase the loss function be accepted. Since there are many local minima, this latter fact is tolerable.

On the other hand, if one believes that there are relatively few local minima, then there is less interest in allowing a $\boldsymbol{\theta}_{new}$ that increases the loss function. Hence, $\tau < 0$, leading to a relatively strict requirement for changing $\boldsymbol{\theta}$ from $\boldsymbol{\theta}_{curr}$. In particular, in the initial (main) decision in step 2, we now only change $\boldsymbol{\theta}_{curr}$ if $y(\boldsymbol{\theta}_{new}) \leq y(\boldsymbol{\theta}_{curr}) + \tau$, where $\tau < 0$. Picking $\tau << 0$ provides inertia to the algorithm by preventing $\boldsymbol{\theta}$ updates unless there is a strong probability that the new $\boldsymbol{\theta}$ improves the loss value. Of course, even with such a conservative approach, the second part of step 2, which is fundamental to SAN, will always provide some chances of accepting a temporarily poorer $\boldsymbol{\theta}$ in the hopes of escaping local minima.

Proposition 8.1 below provides a probabilistic bound on the likelihood of the events $L(\boldsymbol{\theta}_{new}) > L(\boldsymbol{\theta}_{curr})$ and $y(\boldsymbol{\theta}_{new}) \leq y(\boldsymbol{\theta}_{curr}) + \tau$ occurring simultaneously, where τ is in terms of some negative multiple of the standard deviations of the overall measurement noise due to both $\varepsilon(\boldsymbol{\theta}_{new})$ and $\varepsilon(\boldsymbol{\theta}_{curr})$. Hence, the bound concerns the probability of making a "mistake" via having the data indicate an improved (lowered) loss function even though the actual loss increases. When this probability bound is small, there are strong indications that the acceptance of a new value of $\boldsymbol{\theta}$ (i.e., $y(\boldsymbol{\theta}_{new}) \leq y(\boldsymbol{\theta}_{curr}) + \tau$) is associated with the desirable outcome of $L(\boldsymbol{\theta}_{new}) \leq L(\boldsymbol{\theta}_{curr})$. This bound is distribution-free in the sense that it relies on a one-sided version of the Chebyshev inequality (sometimes called the Cantelli inequality). The inequality states that $P(X \geq c) \leq \text{var}(X)/[\text{var}(X) + c^2]$, where X is a mean-zero random variable and $c > 0$ (Tong, 1980, p. 155). (This contrasts with the standard two-sided Chebyshev inequality: $P(|X| \geq c) \leq \text{var}(X)/c^2$.)

Proposition 8.1. Suppose that the measurement noise terms in $y(\boldsymbol{\theta}_{new})$ and $y(\boldsymbol{\theta}_{curr})$, $\varepsilon(\boldsymbol{\theta}_{new})$ and $\varepsilon(\boldsymbol{\theta}_{curr})$, have the same mean and are uncorrelated with finite variances σ^2_{new} and σ^2_{curr}. If $\tau = -c\sqrt{\sigma^2_{new} + \sigma^2_{curr}}$, $c > 0$, then

$$P\left(\{L(\boldsymbol{\theta}_{new}) \geq L(\boldsymbol{\theta}_{curr})\} \cap \{y(\boldsymbol{\theta}_{new}) \leq y(\boldsymbol{\theta}_{curr}) + \tau\}\right) \leq \frac{1}{1 + c^2}. \qquad (8.5)$$

Note. If several, say N_{avg}, values of $y(\cdot)$ are collected at each $\boldsymbol{\theta}$ and averaged for use in the algorithm, then result (8.5) continues to hold provided that τ above is replaced by $\tau = -c\sqrt{(\sigma^2_{\text{new}} + \sigma^2_{\text{curr}})/N_{\text{avg}}}$.

Proof. For convenience, let $\delta L = L(\boldsymbol{\theta}_{\text{new}}) - L(\boldsymbol{\theta}_{\text{curr}})$ and $\delta\varepsilon = \varepsilon(\boldsymbol{\theta}_{\text{new}}) - \varepsilon(\boldsymbol{\theta}_{\text{curr}})$. Then the probability of interest is

$$\begin{aligned} P\big(\{L(\boldsymbol{\theta}_{\text{new}}) \ge L(\boldsymbol{\theta}_{\text{curr}})\} \cap \{y(\boldsymbol{\theta}_{\text{new}}) \le y(\boldsymbol{\theta}_{\text{curr}}) + \tau\}\big) &= P(\delta L \ge 0, \delta L + \delta\varepsilon \le \tau) \\ &= P(0 \le \delta L \le \tau - \delta\varepsilon) \\ &\le P(0 \le \tau - \delta\varepsilon). \end{aligned}$$

Note that $\text{var}(\delta\varepsilon) = \sigma^2_{\text{new}} + \sigma^2_{\text{curr}}$ by the uncorrelatedness assumption. Then, by the above-mentioned one-sided Chebyshev inequality, the last line above satisfies

$$P(\tau - \delta\varepsilon \ge 0) = P(-\delta\varepsilon \ge -\tau) \le \frac{1}{1+c^2},$$

which proves (8.5). ❑

An example implication of Proposition 8.1 is that for sufficiently large c, (8.5) implies that $y(\boldsymbol{\theta}_{\text{new}}) \le y(\boldsymbol{\theta}_{\text{curr}}) + \tau$ is strongly associated with the desirable outcome of $L(\boldsymbol{\theta}_{\text{new}}) < L(\boldsymbol{\theta}_{\text{curr}})$. For instance, if $c = 3$, then the probability is no greater than 0.1 of an incorrect decision in the sense that $L(\boldsymbol{\theta}_{\text{new}}) \ge L(\boldsymbol{\theta}_{\text{curr}})$ happens while $y(\boldsymbol{\theta}_{\text{new}}) \le y(\boldsymbol{\theta}_{\text{curr}}) + \tau$. Given that the bound in (8.5) is built from the Chebyshev inequality (albeit the stronger one-sided version), it is still quite conservative. If, for example, the noises are known to be Gaussian distributed, then the above-mentioned example with $c = 3$ yields a bound to the probability on the left-hand side of (8.5) of only 0.0013, an 80-fold reduction from the distribution-free Chebyshev bound. Exercise 8.2 is aimed at providing additional insight into the distinction between the one- and two-sided (standard) Chebyshev inequality and the Gaussian assumption.

Unfortunately, while the bound in Proposition 8.1 is of some interest in predicting *per-iteration* behavior of SAN, it is unclear what the implication of the bound is for the full range of a search process. Suppose, for example, that the bound tells us that there is likely to be an error no more than 10 percent of the time in the sense of observing $y(\boldsymbol{\theta}_{\text{new}}) \le y(\boldsymbol{\theta}_{\text{curr}}) + \tau$ when $L(\boldsymbol{\theta}_{\text{new}}) \ge L(\boldsymbol{\theta}_{\text{curr}})$ is true. It is unclear what this says about the potential convergence to a global solution $\boldsymbol{\theta}^*$.

8.3 SOME EXAMPLES

This section considers three example problems that illustrate the application of SAN. The first is the famous traveling salesperson problem and the next two are small problems involving a comparison of SAN with two random search methods of Chapter 2 and with a deterministic search method.

Before presenting the first example, let us provide some general background on the traveling salesperson problem. This problem is simple to state, but not necessarily simple to solve. A traveling salesperson must visit every city in some territory once and only once. The problem is to find a minimum cost path meeting this goal when the cost between any pair of cities is known. Because there is a finite number of possible paths, this is a discrete optimization problem, although one with a potentially very large number of elements in Θ. A tour with $n \geq 3$ cities has $(n-1)!/2$ possible unique tours, corresponding to the number of elements in Θ. (*Unique* here refers to fundamentally different ordering. For example, if $n = 3$, the tour 1–2–3–1 is considered equivalent to 1–3–2–1 by the symmetry of the cost between cities, where the indicated numbers 1, 2, or 3 represent the labels for the three cities.) For example, finding an optimal tour of the largest cities in all 50 states of the United States entails 3×10^{62} possible routes, far more than the estimated number of atoms in the Earth ($\sim 10^{50}$). In the language of combinatorial optimization, the problem is NP hard ("NP" variously stands for *nonpolynomial* or *nondeterministic polynomial*; general NP problems are believed to have no solution in computing time proportional to a polynomial of the problem size n—see, e.g., Culberson, 1998).

The perturbations in the SAN algorithm are tied to three fundamental operations on this combinatorial problem: inversion, translation, and switching. Inversion selects two cut points from the current tour and reverses (inverts) the order of the cities between these two cut points. Translation selects a subsection of the current tour and inserts it between two randomly selected other cities. Switching randomly selects two cities and switches their positions on the tour. Figure 8.1 illustrates these operations on an eight-city tour.

Example 8.1—Traveling salesperson problem. One of the most successful applications of SAN has been on this well-known problem. Let us consider a relatively modest-sized problem involving 10 cities. Hence, $p = 10$, with each component of $\boldsymbol{\theta}$ representing one of the city labels 1, 2,..., 10. (SAN has been used successfully in much larger problems—in the sense of providing a much-improved tour—but then it is rarely possible to know "truth" for comparison purposes. For example, the seminal 1983 paper of Kirkpatrick et al. considers a 400-city tour.) The constraint set Θ is such that $\boldsymbol{\theta}$ must represent a valid tour; that is, each of the 10 integers must appear once and only once in $\boldsymbol{\theta}$. (The salesperson can start the trip at the city corresponding to any element of $\boldsymbol{\theta}$; hence, there is no need to constrain the first element to be the origin city.)

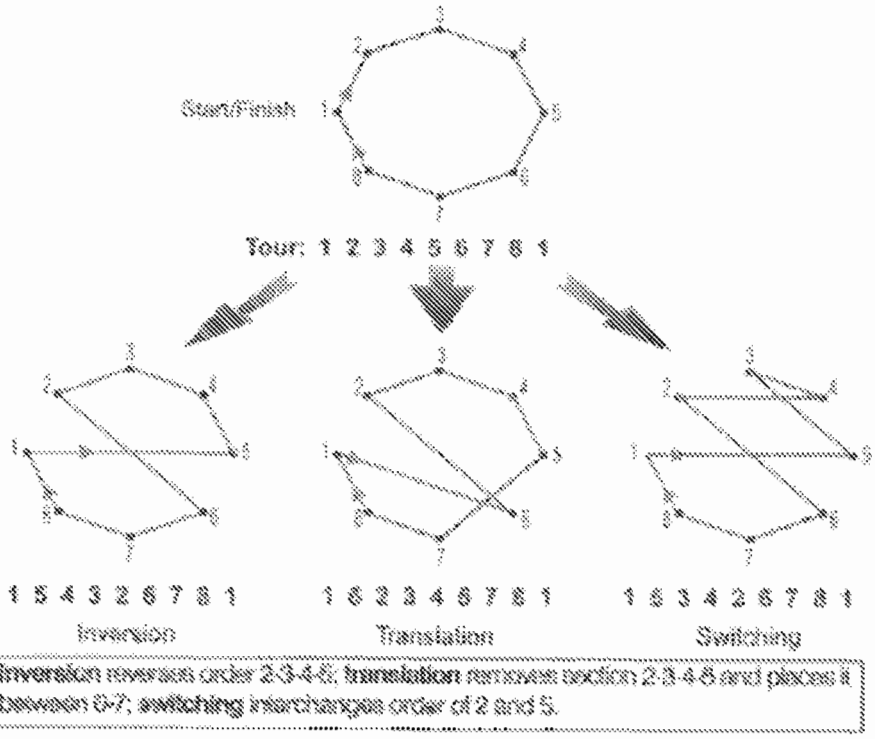

Figure 8.1. Simple network of eight cities to be visited on the tour with the city order listed below each tour. The operations of inversion, translation, and switching are shown relative to the original tour on top.

The city-to-city costs are generated randomly according to a uniform distribution between 10 and 125. Based on the set of randomly generated costs (which are fixed throughout the study), we did an exhaustive search through the $(10-1)!/2 = 181{,}440$ possible tours to determine the optimal tour. The optimal tour has a cost of 440.

Results are determined for 10 independent SAN runs, each initialized at a randomly generated tour and using $N = 8000$ loss evaluations (noise-free in this case). One of the three operations in Figure 8.1 is used to produce each new value $\boldsymbol{\theta}_{new} \in \Theta$ in step 1 of the SAN algorithm. The inversion, translation, and switching operations are selected randomly with probabilities 0.75, 0.125, and 0.125, respectively. A stepwise temperature decay is used, with 40 iterations occurring at each temperature. The initial temperature T is 70 and the decay factor $\lambda = 0.95$.

Table 8.1 shows representative results for five of the 10 replications. The sample mean loss over the 10 replications is 444.1, which is close to the optimum 440 and a significant improvement over the sample mean of the initial loss values of approximately 700. Eight of the 10 runs found the optimal tour (the optimal runs are equivalent versions of the same tour, as shown in the four

Table 8.1. Representative results for five (of 10) SAN runs on the 10-city traveling salesperson problem. Each run uses 8000 loss measurements (7999 iterations).

$\hat{\boldsymbol{\theta}}_{7999}^{T}$ (final 10-city tour)	$L(\hat{\boldsymbol{\theta}}_{7999})$
[1, 5, 2, 4, 7, 6, 8, 9, 3, 10]	440
[5, 1, 10, 7, 4, 2, 6, 8, 9, 3]	459
[1, 10, 3, 9, 8, 6, 7, 4, 2, 5]	440
[2, 5, 1, 10, 3, 9, 8, 6, 7, 4]	440
[9, 3, 10, 1, 5, 2, 4, 7, 6, 8]	440

optimal runs of Table 8.1). An adequate frequency of use for the inversion operator is central to the results here. If the probability of inversion is reduced to 0.50 (versus 0.75), none of the 10 runs yielded an optimal tour. (The relatively greater effectiveness of inversion for traveling salesperson problems with symmetric city-to-city costs is explained in Reidys and Stadler, 2002.) ❑

Let us now evaluate SAN in two small-scale problems with continuous $\boldsymbol{\theta}$. Of course, as emphasized in Section 1.2, one should avoid drawing *general* conclusions from the results of such numerical comparisons. Nevertheless, the comparison here sheds at least limited light on the performance of SAN. Section 8.4 also presents a numerical evaluation of SAN relative to a different annealing-type algorithm introduced in that section; the loss function in the evaluation of Section 8.4 is more complex than the losses here.

Example 8.2—Comparison of random search algorithms and SAN. This example compares the outcomes of random search algorithms B and C (Subsection 2.2.2) with SAN for the case of the simple $p = 2$ quartic polynomial loss function first seen as an example in Section 1.4: $L(\boldsymbol{\theta}) = t_1^4 + t_1^2 + t_1 t_2 + t_2^2$, where $\boldsymbol{\theta} = [t_1, t_2]^T$. Each of the random search and SAN results is based on a $N(0, \rho^2 \boldsymbol{I}_2)$ perturbation distribution (to generate the candidate value of $\boldsymbol{\theta}$); ρ is the same for all SAN implementations (i.e., all initial T), but differs in algorithms B, C, and SAN. Although local minima are not a problem with this simple loss function, this evaluation serves as a test of SAN on a simple function with known solution. Only noise-free loss measurements are considered in the algorithms. Table 8.2 presents the results based on an average of 40 independent replications, with each replication initialized at $\hat{\boldsymbol{\theta}}_0 = [1, 1]^T$. Note that SAN reduces to algorithm B when T is very small (i.e., the Metropolis criterion is then near zero, indicating that $\boldsymbol{\theta}_{\text{new}}$ is almost always rejected unless it produces a

Table 8.2. Comparison of random search and SAN algorithms. Sample mean of terminal $L(\boldsymbol{\theta})$ values after N function evaluations for varying initial temperature T (initial loss value = 4.0).

N	Random Search		SAN		
	Alg. B	Alg. C	Initial T = 0.01	Initial T = 0.10	Initial T = 1.0
100	0.00053	0.328	1.86	0.091	0.763
1000	2.8×10^{-5}	1.1×10^{-5}	0.0092	0.067	0.506
10,000	2.7×10^{-6}	2.5×10^{-7}	0.00038	0.0024	0.018

decrease in L). The results shown for SAN here have T large enough to retain the essential character of the Metropolis decision step.

Reasonable efforts were made to tune the SAN parameters to enhance the performance of the algorithm. A standard stepwise temperature decay is used, with 50 iterations occurring at each temperature and a decay factor $\lambda = 0.98$. Table 8.2 shows the results for several initial temperatures, T.

While SAN produces a marked improvement in the loss function as N increases, the random search techniques tend to produce even better results. So, this example demonstrates that SAN does not always yield performance superior to the simpler random search algorithms. (Exercise 8.6 considers this example further, including a lower initial T. Results for SAN do not significantly improve with a lower initial T.) ❑

Example 8.3—Evaluation of SAN in problem with multiple local minima. There are many numerical studies in the literature showing favorable results for SAN. Let us summarize one on a constrained problem with continuous $\boldsymbol{\theta}$, as described in Brooks and Morgan (1995). The aim of this study is to compare SAN with a popular method of deterministic optimization in a problem with many local minima. Consider $L(\boldsymbol{\theta}) = t_1^2 + 2t_2^2 - 0.3\cos(3\pi t_1) - 0.4\cos(4\pi t_2)$ with $\boldsymbol{\theta} = [t_1, t_2]^T$ and $\Theta = [-1, 1]^2$. This function has many local minima with a unique global minimum at $\boldsymbol{\theta}^* = \mathbf{0}$. This study compares the popular quasi-Newton algorithm from deterministic optimization (mentioned in Section 1.4) with SAN. (The quasi-Newton algorithm has some of the fast convergence properties of the Newton–Raphson algorithm, but is generally more stable in practice; it is one of the most popular nonlinear programming methods.) Note that, in some sense, this comparison is "apples versus oranges" because the quasi-Newton method uses gradient information while SAN uses only loss information.

The quasi-Newton method is run 100,000 times, each with a different initial condition $\hat{\boldsymbol{\theta}}_0 \in \Theta$. Each initial condition is generated randomly

(uniformly) in Θ. It is found that fewer than 20 percent of the quasi-Newton runs yield convergence to points near $\boldsymbol{\theta}^*$. In the other runs, the algorithm settles near one of the local minima. In contrast, all of 1000 runs of SAN with stepwise decaying temperatures settle near $\boldsymbol{\theta}^*$; as with the quasi-Newton evaluation, the initial conditions are generated uniformly for each run. This study shows the relative effectiveness of SAN in finding a global minimum. Additional study details may be found in Brooks and Morgan (1995). ❑

8.4 GLOBAL OPTIMIZATION VIA ANNEALING ALGORITHMS BASED ON STOCHASTIC APPROXIMATION

The general SAN algorithm form discussed in Sections 8.2 and 8.3 is probably what most people involved with stochastic optimization think of when considering annealing algorithms. The basic idea of slow cooling for global optimization, however, can be implemented in other ways as well. One popular way is via connections to the stochastic approximation framework discussed in Chapters 4–7. In particular, let us begin with the basic SA form

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \boldsymbol{G}_k(\hat{\boldsymbol{\theta}}_k), \tag{8.6}$$

where $\boldsymbol{G}_k(\cdot)$ typically represents either a direct (noisy) gradient measurement as in $\partial Q/\partial\boldsymbol{\theta}$ in Chapter 5 or a gradient approximation built from noisy function measurements as in Chapters 6 and 7.

As discussed in Chapters 4–7, the basic algorithm (8.6) is generally for *local* optimization, although one important exception is discussed at the end of this section. To achieve global convergence, the algorithm is modified to include a user (Monte Carlo)-injected random input term $\boldsymbol{w}_k$ scaled by an SA gain coefficient b_k that decays to zero:

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \boldsymbol{G}_k(\hat{\boldsymbol{\theta}}_k) + b_k \boldsymbol{w}_k. \tag{8.7}$$

Typically, $\{\boldsymbol{w}_k\}$ is an independent, identically distributed (i.i.d.) standard Gaussian (i.e., $N(\boldsymbol{0}, \boldsymbol{I}_p)$) sequence, but this distribution is not required. Note that because (8.7) is an SA algorithm, the algorithm is designed to cope with possibly noisy loss or gradient measurements.

Intuitively, one can see how (8.7) offers the possibility of global convergence. Namely, the injected random input $b_k\boldsymbol{w}_k$ provides additional "bounce" to the iteration process and helps avoid premature entrapment in a local minimum. This is analogous to the decision step 2 in the SAN algorithm of Section 8.2, where the algorithm sometimes accepts a poorer value of the loss function in the hopes of leaving a local minimum en route to a global minimum. Further, analogous to SAN, the chances of leaving a minimum as the iteration

process proceeds is reduced to allow for reaching a solution that is a global minimum. This explains the requirement that $b_k \to 0$. To establish some guarantees of global convergence, it is important that b_k decay at a sufficiently slow rate. The intuition behind this should be clear: If $b_k \to 0$ too quickly, then for moderately large k, (8.7) would too closely resemble the local algorithm (8.6) and may not have enough "bounce" to ensure escaping a local minimum.

A number of papers have considered the basic algorithm form (8.7)—or its continuous-time analogue—and have established formal conditions for convergence to a global minimum. Among them are Geman and Hwang (1986), Kushner (1987), Gelfand and Mitter (1991, 1993), Fang et al. (1997), and Yin (1999). There is no one standard set of conditions on the gains a_k and b_k in contrast to the reasonably standard conditions on a_k (and, if relevant, c_k) for the local SA algorithms discussed in Chapters 4–7. We list below four different conditions on the gains and the references from which they are drawn. It is assumed that $a > 0$, $b > 0$, $A \geq 0$, and $B > 0$.

Four Possible Gain Conditions for Global Convergence of (8.7)

A. $a_k = a/\log(k+1)$, $b_k = a_k$ (Kushner, 1987).

B. $a_k = a/(k+1+A)$, $b_k = b/\sqrt{(k+1)\log\log(k+1)}$ (Gelfand and Mitter, 1991, 1993).

C. $a_k = a/(k+1+A)^{\alpha}$, $b_k = b/[(k+1)^{\alpha/2}\log(k+1)]$, $0 < \alpha < 1$ (Fang et al., 1997).

D. $a_k = a/(k+1+A)^{\alpha}$, $b_k = b/\sqrt{(k+1)^{\alpha}\log[(k+1)^{1-\alpha}+B]}$, $0 < \alpha < 1$ (Yin, 1999).

While conditions A and B are used exclusively in their respective references, condition C is only one example from a more general set of conditions on a_k and b_k in Fang et al. (1997). Condition D is one of two possible cases (the other pertains to the classical $\alpha = 1$ setting). Using these gain conditions together with other conditions on $L(\boldsymbol{\theta})$, the random noise $\varepsilon(\boldsymbol{\theta})$ or $e(\boldsymbol{\theta})$ (the difference between $y(\boldsymbol{\theta})$ and $L(\boldsymbol{\theta})$ in the gradient-free case; the difference between $\partial Q/\partial\boldsymbol{\theta}$ and $\boldsymbol{g}(\boldsymbol{\theta})$ in the gradient-based case), and the random input $\boldsymbol{w}_k$, the authors are able to establish various forms of convergence of $\hat{\boldsymbol{\theta}}_k$ to the global minimum of $L(\boldsymbol{\theta})$ (or, more generally, to an element in the *set* of global solutions if more than one global minimum exists). Gelfand and Mitter (1991, 1993), for example, establish convergence *in probability* (versus the stronger almost sure convergence for the local SA algorithms).

As with the local SA algorithms, it is of interest to analyze the stochastic *rate* at which $\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^* \to \mathbf{0}$ for (8.7). Yin (1999) finds that this rate of convergence is $O(1/\sqrt{\log k})$ when using the gain combination in D above and

$O(1/\sqrt{\log\log k})$ when using $a_k = a/(k+1)$ and a companion b_k that is analogous to the b_k in condition D. Both of these rates are painfully slow. The convergence rates found by Yin (1999) are typical of the rates for the global SA algorithm in (8.7); similar rates are available, for example, in Gelfand and Mitter (1993). The slow rates of convergence suggest that practical global convergence—at least via algorithms of the form (8.7)—may sometimes be enhanced through the use of a local algorithm (i.e., no injected randomness) once there is evidence that the iterate is near the solution. (Recall that the local algorithm with noisy gradient or loss measurements has the faster rate $O(1/k^{\eta})$, $0 < \eta \leq 1/2$ or $0 < \eta \leq 1/3$, as discussed in Chapters 4 and 7, respectively.) Note, however, that the slow global rates are asymptotically based; in practical problems, acceptable finite-sample accuracy may often be available with (8.7) directly (see Example 8.4).

The example below evaluates the performance of the SA-based annealing algorithm in (8.7) relative to the SAN algorithm of Section 8.2. This problem is one with a challenging loss function having a large number of local minima (and one global minimum).

Example 8.4—Comparison of global SA in (8.7) and SAN on problem with multiple minima. Given that only noisy loss measurements are available to the algorithms, this example compares global SA and SAN on a challenging problem with many local minima. The global SA algorithm is implemented here using only function evaluations in forming $\boldsymbol{G}_k(\cdot)$ so that we can fairly compare algorithms for a common number of loss functions (versus, say, using the stochastic gradient-based algorithm). In particular, for $\boldsymbol{G}_k(\cdot)$ we use the simultaneous perturbation SA (SPSA) gradient approximation with Bernoulli random perturbations based on the gain sequence combination in item C above (Fang et al., 1997). The SAN algorithm has the standard stepwise decaying temperature form. Both algorithms are run with 20,000 loss measurements per replication. This study considers the loss function:

$$L(\boldsymbol{\theta}) = 0.05\sum_{i=1}^{p} t_i^2 - 40\prod_{i=1}^{p}\cos(t_i), \quad \boldsymbol{\theta} = [t_1, t_2, \ldots, t_p]^T, \tag{8.8}$$

where $p = 10$. This loss has one global minimum at $\boldsymbol{\theta}^* = \mathbf{0}$ and a large number of local minima (Styblinski and Tang, 1990, Example 6). The initial condition $\hat{\boldsymbol{\theta}}_0$ is taken at a local minimum $6.2676[1, 1, \ldots, 1]^T$ (not *quite* a multiple of π = 3.14159...). It is assumed that only noisy loss measurements are available, with the noise being i.i.d. $N(0, 1)$.

As with Tables 8.1 and 8.2, efforts are made to tune the algorithm coefficients to achieve approximately optimal performance. The specific values for the gain sequence in global SA (form C above) are $a = 20$, $A = 100$, $b = 0.5$, $c = 10$, and, as recommended in Chapters 6 and 7, $\alpha = 0.602$ and $\gamma = 0.101$. The SAN runs use an initial $T = 20$ with 100 iterations at each temperature; the

temperature decay factor is $\lambda = 0.95$. To accommodate the noise, the implementation uses a combination of averaging of loss functions ($N_{avg} = 2$) and modification to the decision criterion ($\tau = 1.0$) (see Subsection 8.2.2).

Table 8.3 reports on the results of five independent replications for each of global SA and SAN (the algorithms in each of the five pairs are initialized at the same random number seeds). Some interesting patterns emerged from the study. First, the table indicates superior performance for global SA. In the initial tuning of the algorithm coefficients, the use of short (efficient) realizations in global SA provided a good indication of gains that would work well for the longer ($N = 20{,}000$) realization of actual interest. For SAN, this was not the case, as results for short realizations gave little clue about good coefficient values in the realizations based on $N = 20{,}000$. Although poorer than the global SA results, these SAN results are an improvement over those given in Styblinski and Tang (1990), which relied on $N = 50{,}000$ *noise-free* measurements and used a different implementation of the basic SAN steps in Section 8.2. In particular, Styblinski and Tang (1990) use the so-called fast simulated annealing algorithm of Szu and Hartley (1987), which employs an iteration-by-iteration decay of temperature and Cauchy-distributed perturbations to generate the new (candidate) $\boldsymbol{\theta}$ values. ❑

SA with injected randomness (à la (8.7)) is not the only way in which SA and annealing are combined to achieve global optimization. Section 7.7 mentioned that basic SPSA can often be used for global optimization as a result of the effective noise introduced through the "sloppy" gradient approximation. That is, the SPSA recursion can be expressed as

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \boldsymbol{g}(\hat{\boldsymbol{\theta}}_k) + a_k \text{error}_{\text{noise}} + a_k \text{error}_{\text{perturbation}}, \tag{8.9}$$

Table 8.3. Terminal L value for global SA and SAN with noisy loss measurements and N = 20,000 function evaluations per replication ($L(\hat{\boldsymbol{\theta}}_0) = -20.3$; $L(\boldsymbol{\theta}^*) = -40.0$)).

Replication	Global SA	SAN
1	–39.2	–29.1
2	–39.2	–32.8
3	–39.5	–28.0
4	–39.2	–29.1
5	–38.6	–31.6
Sample Mean	–39.1	–30.1

where $\text{error}_{\text{noise}}$ is the difference from the true gradient $g(\cdot)$ due to the noise in the loss measurements and $\text{error}_{\text{perturbation}}$ is the difference due to the simultaneous perturbation aspect (which exists even if there are noise-free loss measurements). Maryak and Chin (2001) make the important discovery that the term $a_k \text{error}_{\text{perturbation}}$ on the right-hand side of (8.9) acts in the same statistical way as the Monte Carlo injected randomness $b_k w_k$ on the right-hand side of (8.7). Hence, the basic SPSA provides the needed injected randomness "for free," resulting in global convergence. (While the gain selection guidelines of Subsection 7.5.2 may still have some value, it is generally the case that c should not be near zero to avoid premature convergence. This applies even with noise-free loss measurements.)

There are two important advantages of the global convergence result for basic SPSA: (i) The asymptotic rate of convergence has the faster $O(1/k^{\eta})$ form, $0 < \eta \leq 1/3$ (versus the much slower forms mentioned above for (8.7)) and (ii) there are fewer algorithm coefficients to determine since there is no b_k sequence. Nonetheless, as with any theory, there are restrictions to this result. In general, the injected randomness implementation (8.7) has broader applicability for global optimization. Nevertheless, the basic SPSA result may be very effective in some challenging global optimization problems. For example, in 10 of 10 replications with 2500 noise-free loss evaluations per replication, basic SPSA achieves convergence to θ values having identical optimal losses of –40.0 with the challenging multimodal loss function in (8.8) (Maryak and Chin, 2001).

8.5 CONCLUDING REMARKS

This has been the first of three chapters that explore algorithms with a connection to physical processes. In the case here, the physical process is the cooling of materials via annealing (the next two chapters consider analogies to evolutionary biology).

This chapter considered two broad approaches under the annealing rubric. Most of the discussion focused on the popular simulated annealing algorithm. SAN has become a popular global optimizer for both discrete and continuous optimization problems. It has long been known in metallurgy and other fields that annealing—that is, slowly cooling a substance—provides a crystalline structure with maximum strength. SAN is built on mathematical analogies to this cooling process for physical systems. The Metropolis accept/reject criterion provides a critical part in the SAN algorithm. We also see in Chapter 16 an important application of this criterion in the Markov chain Monte Carlo methods for random number generation and estimation.

The second broad annealing approach is tied to stochastic approximation. SA can be used as a global optimizer with the addition of a Monte Carlo injected random term on the right-hand side of the standard recursion. This injected randomness is annealed (damped) in a manner reminiscent of the temperature

cooling in SAN. The injected randomness provides enough "jump" in the algorithm to ensure global convergence (under the appropriate conditions, of course). We also briefly discussed the use of basic SPSA (Chapter 7) as a global optimizer. This algorithm has a built-in random error term that automatically provides a type of injected randomness.

Markov chain theory can be used to establish formal convergence results for the SAN algorithm in discrete (combinatorial) optimization. Analogously, SA theory can be used in the continuous case (see the chapter appendix, Section 8.6). In particular, one can show that special cases of the SA algorithm with injected noise are effectively equivalent to particular SAN algorithms.

As with other approaches, numerical experiments show a variety of results, some very successful and some less so. Broadly speaking, it appears that SAN may be worth trying in discrete multimodal problems with noise-free loss measurements. Of course, SAN may also be successful in other problems as well. In problems with noisy loss measurements, SA algorithms with or without injected randomness may be preferable. Relative to SAN, SA has a stronger theory for convergence with noisy measurements.

8.6 APPENDIX: CONVERGENCE THEORY FOR SIMULATED ANNEALING BASED ON STOCHASTIC APPROXIMATION[2]

Let us begin by summarizing two relatively early works related to convergence of SAN. These are Geman and Geman (1984) and Geman and Hwang (1986). Geman and Geman (1984) go beyond the informal annealing guidelines of Kirkpatrick et al. (1983) and establish conditions on the annealing schedule for formal convergence of an algorithm similar to the standard SAN discussed here; the algorithm is directly motivated by applications to image restoration (see Section 5.4). The temperature variable in Geman and Geman (1984) controls the shape of the distribution of the pixel intensities (rather than the threshold size and perturbation magnitude of the standard SAN), making it more peaked as time proceeds.

More generally, Geman and Hwang (1986) present a framework for continuous-valued θ optimization. Their approach to optimizing continuous θ is based on a *continuous-time* stochastic process with random input governed by a continuous-time probability distribution. This continuous-time process is represented as an approximation to SAN via a stochastic differential equation. The random input is a *Brownian motion* process, which is a type of continuous-time representation of a Gaussian process. Geman and Hwang (1986) show that the probability distribution describing the solution to this differential equation will, in the limit, be a uniform distribution (continuous or discrete) on the set Θ^* of global minima to $L(\theta)$. In the case of a unique θ^*, this corresponds to

[2]This appendix may be skipped or deferred without causing great harm to the overall understanding of implementation aspects of SAN.

convergence in probability (pr.) to $\boldsymbol{\theta}^*$. While there is a strong connection of this analysis to SAN, there is a slight gap between this approach based on continuous-time processes and the standard SAN algorithm, which is based on *discrete* update steps (see Section 8.2). The convergence theory summarized below is more directly tied to the basic SAN form of Section 8.2.

Let us now present a formal convergence analysis for the Metropolis-based SAN algorithm described in Section 8.2, where we let the iteration-dependent decaying temperature sequence $T = T_k$ depend on the current estimate of $\boldsymbol{\theta}$. As we will see, however, this dependence disappears for large k, making the convergence result effectively applicable to a standard state-independent temperature decaying at a (slow) rate T_k that is proportional to $1/\log\log k$. This result relies on a direct correspondence of the SA algorithm (8.7) to the SAN algorithm when $\boldsymbol{\theta}$ is continuous-valued and L is differentiable. The discussion below is based on Gelfand and Mitter (1993).

In showing the correspondence between SA with injected randomness and SAN, assume that the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ exists on $\mathbb{R}^p$ (it does not actually have to be computed). As usual, suppose that $\hat{\boldsymbol{\theta}}_k$ is the Metropolis-type SAN sequence for optimizing L, with k the iteration counter. To define this sequence in terms of the SA recursion (8.7), let $q_k(\boldsymbol{x}, \boldsymbol{y})$ be a normal density function with mean $\boldsymbol{x}$ and covariance matrix $b_k^2\sigma_k^2(\boldsymbol{x})\boldsymbol{I}_p$, where $\sigma_k(\boldsymbol{x}) = \max\{1, a_k^{\tau}\|\boldsymbol{x}\|\}$ and $0 < \tau < 1/4$ ($\boldsymbol{y}$ is the dummy variable for the density). Further, let $s_k(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-[L(\boldsymbol{y}) - L(\boldsymbol{x})]/T_k\}$ if $L(\boldsymbol{y}) > L(\boldsymbol{x})$, and $s_k(\boldsymbol{x}, \boldsymbol{y}) = 1$ otherwise, where $T_k(\boldsymbol{x}) = b_k^2\sigma_k^2(\boldsymbol{x})/(2a_k)$. (As we see below, $\boldsymbol{x}$ and $\boldsymbol{y}$ represent various values of $\boldsymbol{\theta}$.) The function $s_k(\boldsymbol{x}, \boldsymbol{y})$ acts as the *acceptance probability* (analogous to step 2 of the basic SAN algorithm in Section 8.2).

With the appropriate interpretation, the above formulation fits directly into the SAN steps of Section 8.2. The SAN sequence can be obtained through simulation, in a manner similar to the discrete $\boldsymbol{\theta}$ case. Given the current state $\hat{\boldsymbol{\theta}}_k$, generate a candidate solution $\boldsymbol{\theta}_{\text{new}}$ according to (the one-step Markov transition) probability density $q_k(\hat{\boldsymbol{\theta}}_k, \cdot)$. Set the next state $\hat{\boldsymbol{\theta}}_{k+1} = \boldsymbol{\theta}_{\text{new}}$ if $s_k(\hat{\boldsymbol{\theta}}_k, \boldsymbol{\theta}_{\text{new}}) > U_k$, where U_k is an independent $U(0, 1)$ random variable; otherwise, $\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k$. For any set $S \subseteq \mathbb{R}^p$, the sequence $\hat{\boldsymbol{\theta}}_k$ has Markov transition probabilities

$$P(\hat{\boldsymbol{\theta}}_{k+1} \in S \mid \hat{\boldsymbol{\theta}}_k = \boldsymbol{x}) = \int_S q_k(\boldsymbol{x}, \boldsymbol{y}) s_k(\boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} + r_k(\boldsymbol{x}) I_{\{\boldsymbol{x} \in S\}}, \tag{8.10}$$

where $I_{\{\cdot\}}$ is the indicator function for the set $\{\cdot\}$ (1 if $\{\cdot\}$ is true; 0 otherwise) and $r_k(\boldsymbol{x})$ is the normalization $1 - \int_{\mathbb{R}^p} q_k(\boldsymbol{x}, \boldsymbol{y}) s_k(\boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y}$ (see Exercise 8.13 for an equivalent form). Let $\{\boldsymbol{w}_k\}$ be an i.i.d. sequence of $N(\boldsymbol{0}, \boldsymbol{I}_p)$-distributed random vectors and a_k and b_k be suitably chosen sequences.

Following the definition of a noisy root-finding function in Section 1.1 (with noise vector $\boldsymbol{e}_k$), the effective noises $\boldsymbol{e}_k$ here capture the outcome of the Metropolis sampling and decision process. Let $\boldsymbol{e}_k$ be defined by setting

$$\boldsymbol{e}_k = -\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k) + \frac{b_k}{a_k}[1 - \sigma_k(\hat{\boldsymbol{\theta}}_k)d_k]\boldsymbol{w}_k,$$

where the (random) binary decision variable $d_k = 0$ or 1 according to $P(d_k = 1 | \hat{\boldsymbol{\theta}}_k) = s_k(\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\theta}}_k + b_k\sigma_k(\hat{\boldsymbol{\theta}}_k)\boldsymbol{w}_k)$ and it is assumed that $a_k > 0$ for all k (a standard SA assumption, of course). Given the above definitions, the basic SAN steps can be expressed in the SA-type recursion

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k[\boldsymbol{g}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{e}_k] + b_k\boldsymbol{w}_k. \tag{8.11}$$

(Unlike some SA implementations, note that $\boldsymbol{w}_k$ and $\boldsymbol{e}_k$ are *dependent* for all k in this implementation.)

Gelfand and Mitter (1993) use the SA recursion in (8.11) to show that $\hat{\boldsymbol{\theta}}_k$ converges in probability to the set of global minima Θ^* provided that the initial condition is drawn from an appropriate set near the points in Θ^*. A special case of this result applies when L has a unique minimum at $\boldsymbol{\theta}^*$. Then, $P(\|\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\| \geq \eta) \to 0$ as $k \to \infty$ for any $\eta > 0$. That is, the SAN algorithm converges in pr. to $\boldsymbol{\theta}^*$ provided that the a_k and b_k in the temperature sequence $T_k(\hat{\boldsymbol{\theta}}_k) = b_k^2\sigma_k^2(\hat{\boldsymbol{\theta}}_k)/(2a_k)$ are chosen appropriately. Example values of a_k and b_k that are sufficient for convergence are $a_k = a/(k+1)$ and $b_k = b/\sqrt{(k+1)\log\log(k+1)}$, as discussed in Section 8.4.

Although the above convergence result applies for the indicated *state-dependent* temperature sequence, observe that $\sigma_k^2(\boldsymbol{\theta}) = 1$ for all k sufficiently large when $\boldsymbol{\theta}$ lies in a bounded set Θ. So, the dependence of $\sigma_k^2(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$ disappears for large k. In this sense, (8.11) is equivalent to classical SAN for large enough k. That is, the above convergence result is for an algorithm closely related to classical SAN where the temperatures have the *state-independent* values, $T_k = b^2/[2a\log\log(k+1)]$.

EXERCISES

8.1 Using the transition matrix in (8.4), complete the argument showing that repeated application of (8.2) will lead to a system whose state probability converges to the Boltzmann–Gibbs distribution in (8.1) at a fixed temperature T. That is, show $\bar{p}_i = c_T \exp[-x_i/(c_b T)]$ for all i.

8.2 Consider the setting of Proposition 8.1. For $c = 1$, 2, and 5, compare the bound in (8.5) with an analogous result based on the standard (two-sided)

Chebyshev inequality (see Exercise C.4 of Appendix C) and an assumption that the noises are normally distributed. In particular, produce a table contrasting the three bounds for the three values of c. Comment on whether there is any valid reason to use the two-sided Chebyshev inequality in this setting and discuss the benefits of making the Gaussian assumption (if valid!).

8.3 For a 10-city tour in the traveling salesperson problem, illustrate (analogous to Figure 8.1) the inversion of cities 4–8, the translation of cities 4–8 between cities 9 and 10, and the switching of cities 3 and 7.

8.4 With the same traveling salesperson setup and SAN coefficients as in Example 8.1, generate a random city-to-city cost matrix (each element being between 10 and 125). For this cost matrix, determine the optimal tour based on exhaustive search. Run SAN for five replications, showing your results as in Table 8.1.

8.5 Consider a traveling salesperson problem where the cost between cities is equal to the Euclidean norm of the distance between the cities. Explain why the optimal solution path cannot include any links that cross each other.

8.6 With the setting of Example 8.2 (indicated loss function, coefficient settings, number of measurements [N = 100, 1000, and 10,000], and number of replications [40]), implement SAN for three initial temperatures: T = 0.001, 0.10, and 10.0 (tune the perturbation standard deviation ρ so that your results for T = 0.10 are similar to those in Table 8.2). Report results for the three values of N and three initial temperatures. Comment on the results relative to the values in Table 8.2.

8.7 Implement SAN and SPSA for the skewed-quartic loss function in Example 6.6 (Section 6.7) with p = 20. Assume that $\hat{\boldsymbol{\theta}}_0 = [1, 1, \ldots, 1]^T$ and a total loss function budget of N = 2000 measurements for each replication of SAN and SPSA. The measurements for the algorithms have the form $y(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \varepsilon$, where ε is i.i.d. $N(0, \sigma^2)$. Use the stepwise decaying temperature for SAN with 50 iterations at each temperature and tune the coefficients for both SAN and SPSA (use the standard practical gains for SPSA, as described in Section 7.5). Based on the sample mean of the losses at the final iterates over 40 independent replications, compare SAN and SPSA for σ = 0, 1, and 10.

8.8 Consider the loss function $L(\boldsymbol{\theta}) = \frac{1}{2}[(t_1^4 - 16t_1^2 + 5t_1) + (t_2^4 - 16t_2^2 + 5t_2)]$, $\boldsymbol{\theta} = [t_1, t_2]^T$, as given in Styblinski and Tang (1990, Example 1) (also Example 2.3 here).

(a) Identify the four local minima via any convenient algebraic or numerical means.

(b) With a starting $\boldsymbol{\theta}$ of $[4.0, 6.4]^T$, implement the SAN algorithm with stepwise decaying temperatures to find the global minimum (tune the SAN coefficients as desired). Use a maximum of 5000 function evaluations per replication and perform the experiment for five independent replications. Report the five solutions obtained (terminal $\boldsymbol{\theta}$ and $L(\boldsymbol{\theta})$) and the values of the various SAN coefficients.

8.9 Perform Exercise 8.8(b) with independent $N(0, 4^2)$ noise added to the loss function measurements. Use the techniques of Subsection 8.2.2 as appropriate. If you also did Exercise 8.8, how much degradation is seen in the five solutions?

8.10 Consider a $p = 5$ version of the loss function in eqn. (8.8). Identify one local minimum different from 6.2676 $[1, 1, 1, 1, 1]^T$ and not equal to the global minimum. Run SAN five different times using a maximum of 40,000 function evaluations per SAN replication with the identified local minimum as the initial condition. Report the five solutions.

8.11 Consider the four pairs of gains a_k, b_k in Section 8.4 that are valid for global convergence of SA with injected randomness. For $a = b = A = B = 1$ and $\alpha = 0.8$, contrast the four pairs of gains at $k = 1000$ and $k = 1{,}000{,}000$. Comment on reasons for the observed differences in magnitudes across the four combinations.

8.12 Consider the loss function $L(\boldsymbol{\theta}) = -\cos(t_1 - 100)\cos[(t_2 - 100)/\sqrt{2}] + [(t_1 - 100)^2 + (t_2 - 100)^2]/4000$, $\boldsymbol{\theta} = [t_1, t_2]^T$ (sometimes called the *Griewank function*). Let $\Theta = [-200, 400]^2$. The unique global minimum is $\boldsymbol{\theta}^* = [100, 100]^T$.

(a) Analytically or by plotting, provide some sense of roughly how many local minima there are in Θ and specifically identify three of these local minima (other than $\boldsymbol{\theta}^*$).

(b) Based on 50 random initial conditions generated uniformly in Θ, perform 50 replications of SPSA (without the injected noise shown in (8.7)) using 100,000 loss evaluations per replication and the algorithm steps of Subsection 7.5.1 (this implementation is consistent with the discussion at the end of Section 8.4). Assume noise-free loss measurements and use the same gain sequences (a_k and c_k) for all 50 runs. How many of the 50 terminal iterates $\hat{\boldsymbol{\theta}}_{50,000}$ effectively converge to the global minimum in the sense that $\left|L(\hat{\boldsymbol{\theta}}_{50,000}) - L(\boldsymbol{\theta}^*)\right| \le 0.01$ *and* $\left\|\hat{\boldsymbol{\theta}}_{50,000} - \boldsymbol{\theta}^*\right\| \le 0.2$?

8.13 Show that the right-hand side of (8.10) can be written as $\left\{1 - \int_{S^c} q_k(\boldsymbol{x}, \boldsymbol{y}) s_k(\boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \text{ if } \boldsymbol{x} \in S;\ \int_S q_k(\boldsymbol{x}, \boldsymbol{y}) s_k(\boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \text{ if } \boldsymbol{x} \notin S\right\}$.