%A L. Ingber
%T Simulated annealing: Practice versus theory
%J Mathl. Comput. Modelling
%V 18
%N 11
%D 1993
%P 29-57

Simulated annealing: Practice versus theory

Lester Ingber
Lester Ingber Research, P.O.B. 857, McLean, VA 22101
ingber@alumni.caltech.edu

Simulated annealing (SA) presents an optimization technique with several striking positive and negative features. Perhaps its most salient feature, statistically promising to deliver an optimal solution, in current practice is often spurned to use instead modified faster algorithms, “simulated quenching” (SQ). Using the author’s Adaptive Simulated Annealing (ASA) code, some examples are given which demonstrate how SQ can be much faster than SA without sacrificing accuracy.

Keywords: Simulated annealing, random algorithm, optimization technique

1. Introduction

1.1. Shades of simulated annealing

Simulated annealing presents an optimization technique that can: (a) process cost functions possessing quite arbitrary degrees of nonlinearities, discontinuities, and stochasticity; (b) process quite arbitrary boundary conditions and constraints imposed on these cost functions; (c) be implemented quite easily with the degree of coding quite minimal relative to other nonlinear optimization algorithms; (d) statistically guarantee finding an optimal solution. Section 2 gives a short introduction to SA, emphasizing its property of (weak) ergodicity. Note that for very large systems, ergodicity is not an entirely rigorous concept when faced with the real task of its computation [1]. Moreover, in this paper "ergodic" is used in a very weak sense, as it is not proposed, theoretically or practically, that all states of the system are actually to be visited.

Even "standard" SA is not without its critics. Some negative features of SA are that it can: (A) be quite time-consuming to find an optimal fit, especially when using the "standard" Boltzmann technique; (B) be difficult to fine tune to specific problems, relative to some other fitting techniques; (C) suffer from "over-hype" and faddish misuse, leading to misinterpretation of results; (D) lose the ergodic property (d) by misuse, e.g., by transforming SA into a method of "simulated quenching" (SQ) for which there is no statistical guarantee of finding an optimal solution. Section 3 presents some examples to demonstrate how SQ can give misleading results. There also is a large and growing domain of SA-like techniques, which do not theoretically predict general statistical optimality, but which are extremely powerful for certain classes of problems. Section 3 includes some of these algorithms.

Section 4 gives a short description of a sampling of the many complex problems which have benefited greatly by the use of SA and SQ. Specific examples are given from papers addressing robust problems across many disciplines. There are many reviews of simulated annealing, comparisons among simulated annealing algorithms, and between simulated annealing and other algorithms [2-5]. This paper is not as exhaustive as these other reviews were in their time. The sampling presented here is not meant to be a review of SA, but rather a documented statement of the widespread use of SA and SQ. The emphasis is on comparing the basic theoretic constraints of true simulated annealing (SA) with actual practice on a range of problems spanning many disciplines. On one hand, this may help to address what may yet be expected in terms of better necessary conditions on SA to make it a more efficient algorithm, as many believe that the present sufficiency conditions are overly restrictive. On the other hand, perhaps some of the results not adhering to the present sufficiency conditions that are being reported in the literature are quite biased, perhaps being too positive or too negative. An attempt has been made to limit technical discussion to only that necessary to highlight particular approaches.

There are several approaches being researched to develop better SA algorithms and auxiliary algorithms to predict the efficiency of SA on particular problems. These give some insight into how SA might be developed into a faster but still optimal algorithm for many kinds of systems. Section 5 describes some of these approaches.

In Section 6 the author's publicly available code, Adaptive Simulated Annealing (ASA) [6], illustrates how SQ can indeed sometimes perform much faster than SA, without sacrificing accuracy.

This paper appreciates the utility of SQ as a trade-off to benefit from (a), (b) and (c) at the expense of (D). The conclusion, Section 7, iterates the theme in this introduction, of the questionable push to neglect some of the theoretical strengths of SA in favor of expediency, and of some new developments that may make some of these compromises less necessary.

1.2. Critics of SA

At the outset it must be stated that SA is not without its critics. The primary criticism is that it is too slow; this is partially addressed here by summarizing much work in appropriately adapting SQ to many problems. Another criticism is that it is "overkill" for many of the problems on which it is used; this is partially addressed here by summarizing much work demonstrating that it is not insignificant that many researchers are using SA/SQ because of the ease in which constraints and complex cost functions can easily be approached and coded.

There is another class of criticisms that the algorithm is too broadly based on physical intuition and is too short on mathematical rigor [7]. In that particular bitter and scathing critique the authors take offense at the lack of reference to other prior work [8], the use of "metaphysical non-mathematical ideas of melting, cooling, and freezing" reference to the physical process of annealing as used to popularize SA [9], and they give their own calculations to demonstrate that SA can be a very poor algorithm to search for global optima in some instances.

That there are undoubtedly other references that should be more regularly referenced is an objective issue that has much merit, with respect to SA as well as to other research projects. The other criticisms may be considered by some to be more subjective, but they are likely no more extreme than the use of SQ to solve for global optima under the protective umbrella of SA.

2. "Standard" simulated annealing (SA)

The Metropolis Monte Carlo integration algorithm [10] was generalized by the Kirkpatrick algorithm to include a temperature schedule for efficient searching [9]. A sufficiency proof was then shown to put an lower bound on that schedule as $1/\log(t)$, where t is an artificial time measure of the annealing schedule [11]. However, independent credit usually goes to several other authors for independently developing the algorithm that is now recognized as simulated annealing [8,12].

2.1. Boltzmann annealing (BA)

Credit for the first simulated annealing is generally recognized as a Monte Carlo importance-sampling technique for doing large-dimensional path integrals arising in statistical physics problems [10]. This method was generalized to fitting non-convex cost-functions arising in a variety of problems, e.g., finding the optimal wiring for a densely wired computer chip [9]. The choices of probability distributions described in this section are generally specified as Boltzmann annealing (BA) [13].

The method of simulated annealing consists of three functional relationships.

1. $g(x)$: Probability density of state-space of D parameters $x = \{x^i; i = 1, D\}$.
2. $h(\Delta E)$: Probability for acceptance of new cost-function given the just previous value.
3. $T(k)$: schedule of "annealing" the "temperature" T in annealing-time steps k, i.e., of changing the volatility or fluctuations of one or both of the two previous probability densities.

The acceptance probability is based on the chances of obtaining a new state with "energy" E_{k+1} relative to a previous state with "energy" E_k,

$$h(\Delta E) = \frac{\exp(-E_{k+1}/T)}{\exp(-E_{k+1}/T) + \exp(-E_k/T)}$$

$$= \frac{1}{1 + \exp(\Delta E/T)}$$

$$\approx \exp(-\Delta E/T) \ , \tag{1}$$

where ΔE represents the "energy" difference between the present and previous values of the energies (considered here as cost functions) appropriate to the physical problem, i.e., $\Delta E = E_{k+1} - E_k$. This essentially is the Boltzmann distribution contributing to the statistical mechanical partition function of the system [14].

This can be described by considering: a set of states labeled by x, each with energy $e(x)$; a set of probability distributions $p(x)$; and the energy distribution per state $d\big(e(x)\big)$, giving an aggregate energy E,

$$\sum_x p(x) d\big(e(x)\big) = E \ . \tag{2}$$

The principle of maximizing the entropy, S,

$$S = -\sum_x p(x) \ln[p(x)/p(\bar{x})] \ , \tag{3}$$

where $\bar{x}$ represents a reference state, using Lagrange multipliers [15] to constrain the energy to average value T, leads to the most likely Gibbs distribution $G(x)$,

$$G(x) = \frac{1}{Z} \exp\bigl(-H(x)/T\bigr) \ , \tag{4}$$

in terms of the normalizing partition function Z, and the Hamiltonian H operator as the "energy" function,

$$Z = \sum_x \exp\bigl(-H(x)/T\bigr) \ . \tag{5}$$

For such distributions of states and acceptance probabilities defined by functions such as $h(\Delta E)$, the equilibrium principle of detailed balance holds. I.e., the distributions of states before, $G(x_k)$, and after, $G(x_{k+1})$, applying the acceptance criteria, $h(\Delta E) = h(E_{k+1} - E_k)$ are the same:

$$G(x_k) h\bigl(\Delta E(x)\bigr) = G(x_{k+1}) \ . \tag{6}$$

This is sufficient to establish that all states of the system can be sampled, in theory. However, the annealing schedule interrupts equilibrium every time the temperature is changed, and so, at best, this must be done carefully and gradually.

An important aspect of the SA algorithm is to pick the ranges of the parameters to be searched. In practice, computation of continuous systems requires some discretization, so without loss of much generality for applications described here, the space will be assumed to be discretized. There are additional constraints that are required when dealing with generating and cost functions with integral values. Many practitioners use novel techniques to narrow the range as the search progresses. For example, based on functional forms derived for many physical systems belonging to the class of Gaussian-Markovian systems, one could choose an algorithm for g,

$$g(\Delta x) = (2\pi T)^{-D/2} \exp[-\Delta x^2/(2T)] \ , \tag{7}$$

where $\Delta x = x - x_0$ is the deviation of x from x_0 (usually taken to be the just-previously chosen point), proportional to a "momentum" variable, and where T is a measure of the fluctuations of the Boltzmann distribution g in the D-dimensional x-space. Given $g(\Delta x)$, it has been proven [11] that it suffices to obtain a global minimum of $E(x)$ if T is selected to be not faster than

$$T(k) = \frac{T_0}{\ln k} \ , \tag{8}$$

with T_0 "large enough."

For the purposes of this paper, a heuristic demonstration follows, to show that Eq. (8) will suffice to give a global minimum of $E(x)$ [13]. In order to statistically assure, i.e., requiring many trials, that any point in x-space can be sampled infinitely often in annealing-time (IOT), it suffices to prove that the products of probabilities of not generating a state x IOT for all annealing-times successive to k_0 yield zero,

$$\prod_{k=k_0}^{\infty} (1 - g_k) = 0 \ . \tag{9}$$

This is equivalent to

$$\sum_{k=k_0}^{\infty} g_k = \infty \ . \tag{10}$$

The problem then reduces to finding $T(k)$ to satisfy Eq. (10).

For BA, if $T(k)$ is selected to be Eq. (8), then Eq. (7) gives

$$\sum_{k=k_0}^{\infty} g_k \geq \sum_{k=k_0}^{\infty} \exp(-\ln k) = \sum_{k=k_0}^{\infty} 1/k = \infty \ . \tag{11}$$

Although there are sound physical principles underlying the choices of Eqs. (7) and (1) [10], it was noted that this method of finding the global minimum in x-space was not limited to physics examples requiring *bona fide* "temperatures" and "energies." Rather, this methodology can be readily extended to any problem for which a reasonable probability density $h(\Delta x)$ can be formulated [9].

3. Simulated quenching (SQ)

Many researchers have found it very attractive to take advantage of the ease of coding and implementing SA, utilizing its ability to handle quite complex cost functions and constraints. However, the long time of execution of standard Boltzmann-type SA has many times driven these projects to utilize a temperature schedule too fast to satisfy the sufficiency conditions required to establish a true (weak) ergodic search. A logarithmic temperature schedule is consistent with the Boltzmann algorithm, e.g., the temperature schedule is taken to be

$$T_k = T_0 \frac{\ln k_0}{\ln k} \ , \tag{12}$$

where T is the "temperature," k is the "time" index of annealing, and k_0 is some starting index. This can be written for large k as

$$\Delta T = -T_0 \frac{\ln k_0 \Delta k}{k(\ln k)^2} \ , \ k \gg 1$$

$$T_{k+1} = T_k - T_0 \frac{\ln k_0}{k(\ln k)^2} \ . \tag{13}$$

However, some researchers using the Boltzmann algorithm use an exponential schedule, e.g.,

$$T_{k+1} = cT_k \ , \ 0 < c < 1$$

$$\frac{\Delta T}{T_k} = (c-1)\Delta k \ , \ k \gg 1$$

$$T_k = T_0 \exp\big((c-1)k\big) \ , \tag{14}$$

with expediency the only reason given. While perhaps someday some less stringent necessary conditions may be developed for the Boltzmann algorithm, this is not now the state of affairs. The question arises, what is the value of this clear misuse of the claim to use SA to help solve these problems/systems? Below, a variant of SA, adaptive simulated annealing (ASA) [6,16], in fact does justify an exponential annealing schedule, but only if a particular distribution is used for the generating function.

In many cases it is clear that the researchers already know quite a bit about their system, and the convenience of the SA algorithm, together with the need for some global search over local optima, makes a strong practical case for the use of SQ. In some of these cases, the researchers have been more diligent with regard to their numerical SQ work, and have compared the efficiency of SQ to some other methods they have tried. Of course, the point must be made that while SA's true strength lies in its ability to statistically deliver a true global optimum, there are no theoretical reasons for assuming it will be more efficient than any other algorithm that also can find this global optimum.

3.1. Genetic algorithms (GA)

As an example of other algorithms competitive with SQ, there is a very popular class of algorithms, genetic algorithms (GA) that has spawned its own culture across many disciplines. While the origins of its development were not to seek optimization *per se* [17,18], there are reasons to consider GA as valid approaches to numerical optimization [19,20]. This has led to some comparisons between GA and SA techniques [21], which currently must be viewed in the context of "judging" these algorithms only specific to the problems/systems being tested. I.e., it should be expected that there are systems for which one of GA or SA will be better suited than the other. While GA does not possess any claim to ergodicity, albeit there is some progress in establishing convergence to some fixed optima [22], features typically addressed by SQ, such as premature global convergence, rapid local convergence, and the handling of constraints, all can be reasonably treated in the framework of GA [19]. GA also is not without its critics with respect to its approach, and examples have been developed to illustrate how simple random mutation may be superior to GA [23].

3.1.1. GA-SA hybrids

Below a hybrid parallelized SA-GA technique, parallel recombinative simulated annealing (PRSA), is reported to be useful to speed up SA under some circumstances [24]. While the actual test cases reported in the PRSA paper used SQ exponential temperature schedules on Boltzmann algorithms, the PRSA method is an alternative method of taking advantage of flexibility in searching the parameter space, e.g., as does ASA. Given the use of true SA temperature schedules in PRSA, the advantages in optimal searching of the parameter space afforded by ASA could reasonably be overshadowed by some advantages offered by GA, e.g., added degrees of parallelism and perhaps less sensitivity to initial conditions. It would be interesting to explore the application of ASA techniques to the processes of crossover and mutation in the GA stages of PRSA.

There have been other successful attempts to create hybrid GA-SA algorithms. In one approach, the authors have given a proof that an equilibrium distribution can be achieved by using a Metropolis-type acceptance rule [25].

3.2. Some problems with SQ

To make the point of how quenching can lead to some problems, consider some graphs from a previous study [21]. Fig. 1 uses f_0, an objective function which contains a very large number of local minima [26], and is very difficult to optimize. Trajectories were developed in an SA study [21] using very fast simulated reannealing (VFSR) [16,27], discussed below as ASA [6], and a standard genetic algorithm generator [28]. The number of local minima is given by $10^{5n} - 1$; when $n = 4$ it contains 10^{20} local minima. (Visiting each minimum for a millisecond would take about the present age of the universe to visit all minima.)

$$f_0(x_1, \ldots, x_n) = \sum_{i=1}^{n} \begin{cases} (t_i \,\mathrm{sgn}\,(z_i) + z_i)^2 c d_i & \text{if } |x_i - z_i| < |t_i| \\ d_i x_i^2 & \text{otherwise} \end{cases} ,$$

$$z_i = \left\lfloor \left| \frac{x_i}{s_i} \right| + 0.49999 \right\rfloor \mathrm{sgn}\,(x_i) s_i \ ,$$

$$s_i = 0.2,\ t_i = 0.05,\ i = 1, n \ ,$$

$$d_i = \{1.0,\ 1000.0,\ 10.0,\ 100.0, \cdots\} \ ,$$

$$c = 0.15 \ ,$$

$$-1000.0 \le x_i \le 1000.0 \ ,\ i = 1, n \ , \tag{15}$$

where s_i, t_i, d_i (repeated in cycles of 4), and c are coefficients defined such that f_0 defines a paraboloid with axis parallel to the coordinates, and a set of holes that increase in depth near the origin.

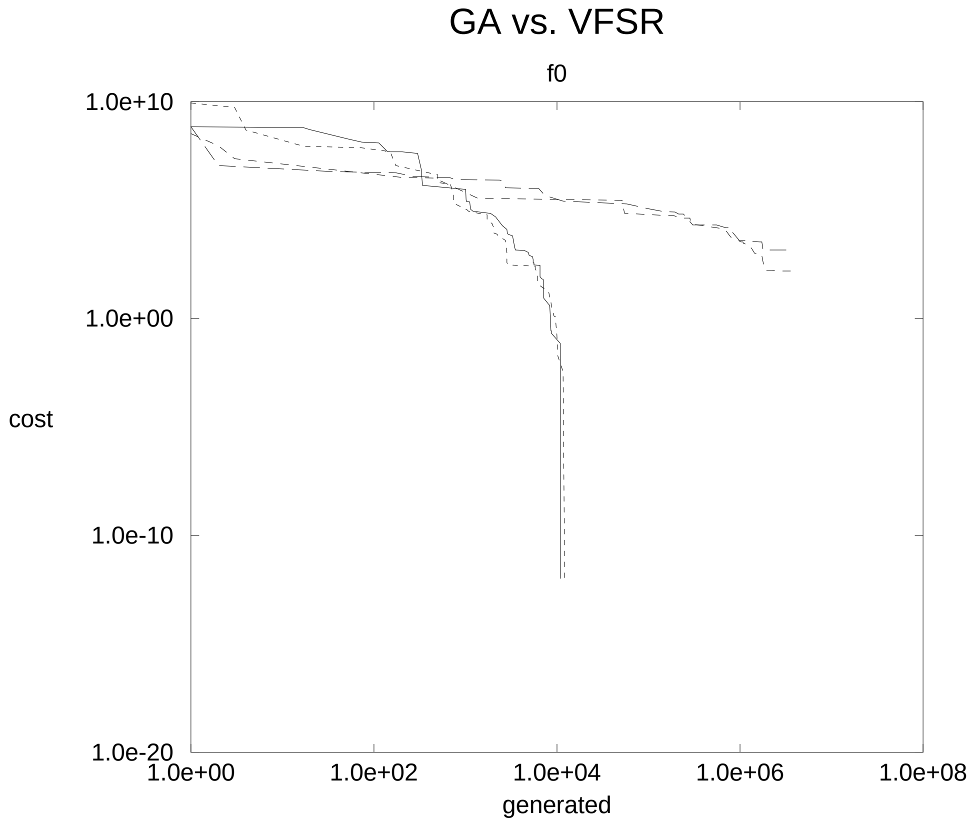

Fig. 1. Comparison between GA and VFSR is given for function f_0, where the dimension of the space is 4. Solid and short dashed lines each represent one VFSR run each, and dashed and long dashed lines represent one GA run each. The runs are log-log plotted to show relative convergence rates of each algorithm. The abscissa indicates the number of function calls, while the ordinate shows the best function evaluation found so far. For purposes of these log-log plots, VFSR was cut off arbitrarily at $f < 10^{-12}$, even when it actually attained 0 to machine precision.

Fig. 2 shows two trajectories when the dimension of f_0 is increased from 4 to 10, presenting a problem with 10^{50} local minima (most of which are beyond a typical workstation's precision and recognition). Clearly, a quenching algorithm might well have not obtained an optimal solution within any practical time. In fact, some standard SA techniques, such as BA and fast annealing (FA, discussed below), can miss global optima as well when optimizing functions with extremely large numbers of minima [29].

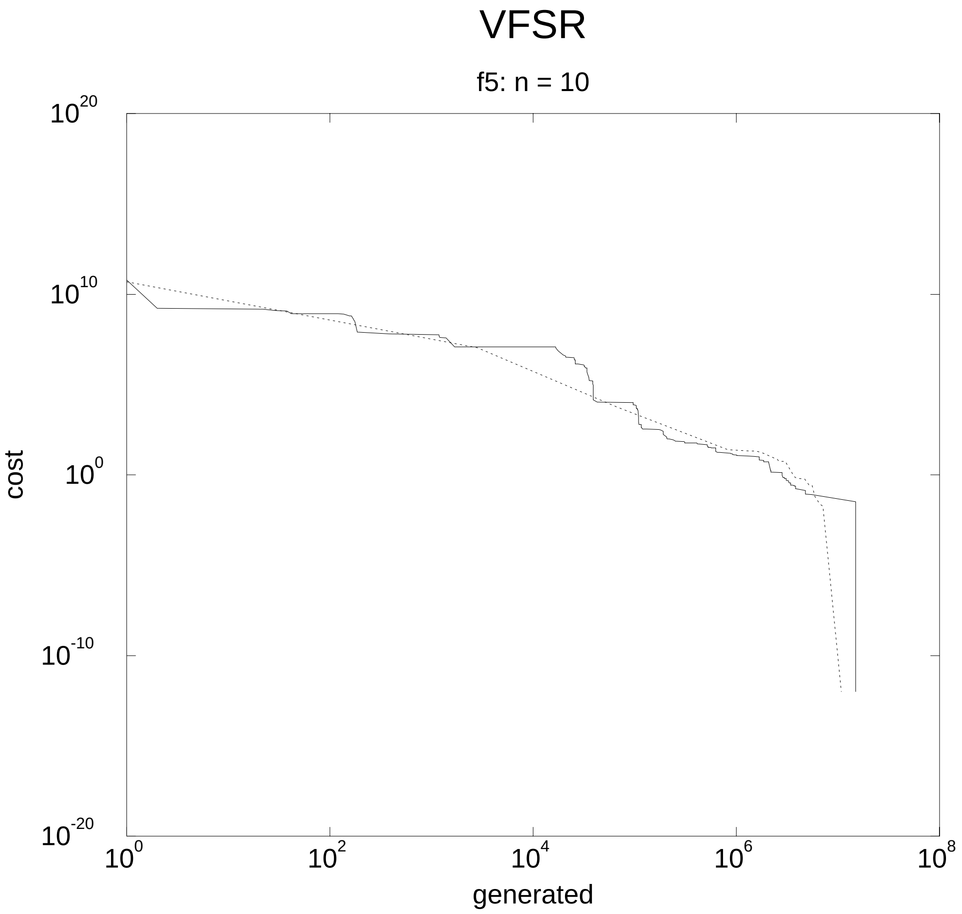

Fig. 2. Trajectories for VFSR are given for function f_0, where the dimension of the space is 10. See Fig. 1 for legend.

Fig. 3 uses f_3, the plateau function, generated as the sum of integer threshold values. The five dimensional space has one minimum and is discontinuous.

$$f_3(x_1, \cdots, x_5) = 30.0 + \sum_{j=1}^{5} \left\lfloor x_j \right\rfloor ,$$

$$-5.12 \le x_i \le 5.12 \;,\; i = 1, 5 \;. \tag{16}$$

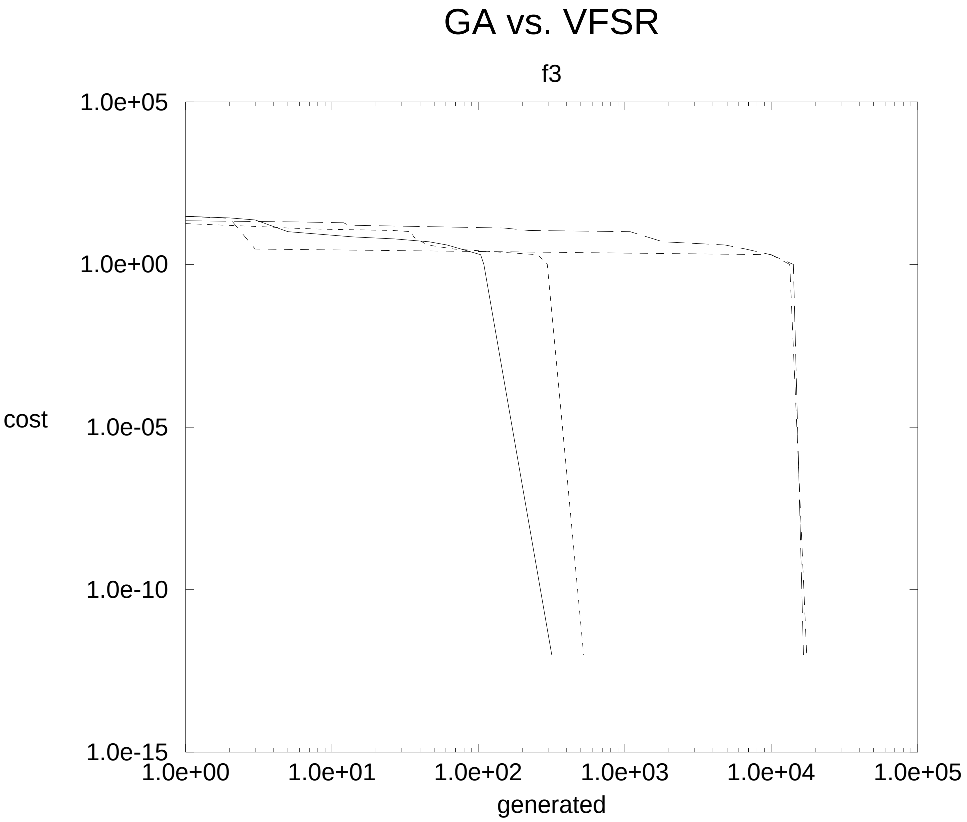

Fig. 3. Comparison between GA and VFSR is given for function f_3. See Fig. 1 for legend.

In Fig. 1, quenching would seem to work quite well if one were using the optimization procedure illustrated by the medium-dashed and long-dashed trajectories, since no clear dramatic benefit seems to be derived by continuing with more detailed searching. However, with respect to the algorithm illustrated by the solid and short-dashed trajectories, especially given no advance knowledge of a given function/data, when should one decide to curtail the search? In this second case, if one does not venture out long enough, the true global minimum will very likely be completely missed!

This point is emphasized again in Fig. 3. If one does not venture out far enough, the global minimum will likely not be reached. Furthermore, here efficiency is irrelevant, since once a favorable approach is determined, the calculation suddenly dives down into the global minimum.

4. Sampling of SA/SQ applications

Because of the very widespread use of simulated annealing over many disciplines, it is convenient to describe a sampling with respect to specific disciplines. A main purpose here is to demonstrate the nontrivial power of SA/SQ to handle quite complex problems/systems and constraints.

4.1. Traveling salesman problem (TSP)

The first popular paper on simulated annealing that drew the attention of many researchers was focussed on optimizing the circuitry of computer chips and on the traveling salesman problem (TSP) [9]. The literature is quite dense with other applications to the TSP, a simple example of an NP-complete problem. The TSP should be included in any list of test problems, if for no other reason than its popularity, but also because it can be considered a prototypical physical model of many quasi-linear systems [30].

In at least one early study, the TSP was used as a test case to try to determine an “efficient” exponential temperature schedule of type Eq. (14), leading to a variant of SQ [31]. In that particular study, advantage was taken of the nature of the TSP and of Boltzmann annealing to test some analytic derivations of expected properties of the algorithm, e.g., of numerical convergence to expected “thermodynamic” properties.

4.2. Circuit design

Applications to more complex circuit design problems including several layers of logic hierarchy were approached using SQ [32]. This required placements and routing for tens to hundreds of groups of units, potentially a higher dimensional task than placing individual connections among units.

While SQ has been effective in determining circuitries, an inverse problem also can be approached. A “Boltzmann machine” SQ algorithm, a variant of mean-field annealing discussed below, was hardwired onto a VLSI chip to perform SQ at very high speeds [33].

4.3. Mathematics/combinatorics

The design of efficient classification and decision trees, an NP-complete problem, greatly benefited by applying SQ, with an exponential temperature schedule

$$T_{i+1} = \alpha T_i \ , \ 0.7 \le \alpha \le 0.99 \ , \tag{17}$$

more so than trying the information-theoretic Huffman algorithm [34].

SQ techniques similarly have been useful in approaching graph problems. In one study, searching for the maximum number of edges in graphs of order $v \le 200$ and girth >= 5, the authors found that their own variant of “hillclimbing” was superior [35]. Another study using the SQ mean-field annealing algorithm (MFA), described below, found SQ and SA superior over other optimization techniques in determining maximal sets of vertices with all pairs connected by an edge [36].

SQ was used to determine subsquare free Latin squares [37]. The authors demonstrated that the ability to recognize an optimal solution made it feasible to use SQ instead of SA.

Mean field annealing (MFA), discussed below, was used to apply neural networks to the minimum cut graph bisection problem, and its speed of solution was found superior to other techniques [38].

Many difficult optimization problems arise concerning matrices. Standard SA was useful in finding optimal block and row-column designs [39]. Another optimization problem used SQ, using low acceptance ratios as the criteria to exit, to optimize row-column permutations designed to diagonalize matrices representing coauthor citation frequencies [40].

4.4. Data analysis

Standard SA was found optimal in some cases, prohibitively slow in others, when applied to exploratory data analysis, i.e., mapping problems of matching distances among patterns in high dimensional spaces and clustering problems in labeling patterns into natural subsets [41].

When looking at controlled rounding procedures in Census data, to preserve the anonymity of respondents, SQ, using an exponential temperature schedule

$$T_j = F T_{j-1} \ ,$$

$$F = (T_{\min}/T_{\max})^{1/N_{cycles}} \tag{18}$$

was found superior, both in speed and in finding optimal solutions, to all other techniques tried [42].

4.5. Imaging

Image reconstruction and filtering requires recognition and extraction of patterns from sets of data. Often, an algebraic model is used to develop a filter to aid in this process. Then, parameters of the model must be fit to data, and here SQ techniques have been quite successful [43,44]. The models often are not very nonlinear, but they are high dimensional.

A very difficult problem, in determining both spatial and temporal aspects of estimation of visual motion over sequences of images, was approached by developing a model invoking continuity of the image with respect to the motion [45]. This gave impetus to develop an SQ algorithm with a non-monotonic temperature schedule that kept lower temperature across fields tracked for longer times, and higher temperatures across newer fields.

Many imaging techniques use techniques of Kalman filtering to extract signal from noise. It has been shown that the Kalman filter can be derived from more general multivariate nonlinear multiplicative-noise systems [46,47]. As such, this presents a more powerful algorithm to model signals and noise. These equations can be represented as nonlinear Lagrangians [48]. SA techniques can be used to fit such models to data, but the author has not yet seen any specific applications of such techniques to imaging.

4.6. Neural networks

There have been several approaches to implementing some form of simulated annealing into neural networks, e.g., to expedite the training phase of fitting parameters. The prototype was the “Boltzmann machine,” [49] which was constructed with many analogies to Ising models of magnetic systems [50].

An SA algorithm, fast annealing (FA) discussed below, was demonstrated to solve the problems of trapping in local optima found with a local gradient technique, when applied to a network using an energy function modeled on the potential of electric charges to study associative recall [51]. The authors found empirically that taking the initial temperature to be the median, instead of the average, of some test random transitions increased the efficiency of their searches. They noted that the Cauchy algorithm used, though theoretically faster than the Boltzmann algorithm, still took quite long to search.

There are other studies that have taken somewhat the opposite approach, that of developing some variants of SQ which possess the same mathematical structure as particular neural networks. Then, those neural networks can be used as a machine to perform the particular variant of SQ. These approaches include mean-field annealing algorithms (MFA) [36,52,53], discussed below. A useful contribution is a method for estimating the critical temperature in advance, obtained by linearizing the mean-field equations and looking for the largest eigenvalue [38]. These neural-network MFA applications have been extended to many other optimization problems, such as TSP, graph bisection and partitioning, scheduling problems, and the knapsack problem with inequality constraints; these are discussed in a review article [54].

Overlapping the category of hardwiring SA/SQ into computer circuitry, there are other algorithms suggested for developing hardware neural network-type implementations of SA/SQ for specific classes of systems. One such application was designed for a Boolean network [55]. Such a machine can be applied to other optimization tasks.

A construction of neural networks applied to image reconstruction has been proposed to utilize a physical implementation of an amorphous magnetic processor [56]. Implicit in their design is the capability for associative memory, fast associative access, and massive information storage. This implementation is based primarily on ideas from the Boltzmann machine, discussed above. An amorphous magnetic material such as a spin glass film is taught a series of pattern vectors by first locally laser heating it in the magnetic field created by an imposed input pattern until the spins become mobile, and then allowing the magnetic spins of the system to seek an energy minimum. After repeated training sessions, an energy landscape is created which effectively stores and permits retrieval of pattern vectors from a set of given inputs.

Another neural network algorithm, based on aggregate columnar circuitries in neocortex, with direct applications to Fokker-Planck/Langevin/Lagrangian systems, was developed by incorporating an SA (ASA) algorithm [57]. The high degree of nonlinearity and stochasticity in such systems make it likely that true SA is required for such problems.

4.7. Biology

The biological sciences are poised for many breakthroughs at many spatial-temporal scales. Biological researchers are bringing to bear modeling techniques from other sciences as well as their own to approach these complex systems. This often requires the fitting of complex cost functions and constraints to experimental data. For example, A very difficult problem in molecular biology concerns understanding the secondary structures of protein chains, e.g., helices, turns, extended molecular strands, etc. Models are being developed that attempt to explain the secondary structures given the primary composition of the chains, usually within a prescribed context/environment, e.g., in the presence of other specific protein chains.

A prototypical problem of finding conformational substates of proteins was mapped onto the TSP, discussed above, but using more realistic van der Waals interactions to determine "distances" between units [58]. Then, an SQ algorithm was used to determine optimal configurations.

One direct approach minimizes the "free energy" (the internal energy minus the temperature times the entropy), finding the conformation with the lowest free energy. SA has been invoked to approach such difficult computational problems, but in practice SQ usually is used. For example, one study [59] uses a Boltzmann algorithm, enhanced with an Acceptance Ratio Method (ARM) using updated information to select step sizes for individual atoms or clusters of atoms, these being ranges of the relevant parameters in the free energy considered as cost function. This method has been shown to be effective in treating such inhomogeneous systems. They pick an exponential SQ temperature schedule

$$T_{i+1} = \gamma T_i \ ,$$

$$\gamma = (T_f / T_i)^{\frac{1}{N_c - 1}} \ , \tag{19}$$

and determine γ from a predetermined number of annealing temperature-cycles, N_c, which establishes a progression from initial temperature T_i to a final temperature T_f.

Another approach to a similar problem develops a model in information-theoretic terms [60]. First, a minimal message length (MML) encoding is used to establish a small set of candidate graphs representing secondary structures. Then, probabilities are attached to each leg of each graph, and an SA algorithm (following close to a $1/\log(t)$ schedule) is used to find the graph with maximum probability and the optimal graph. The combined MML and maximal probability model might be considered as quenching in itself, in that, if an algorithm could be developed to simultaneously optimize the combined problem with SA, a different optimal solution might evolve. This is not a criticism of the the very difficult work done to date, but rather a statement of a generic problem in optimizing a system with a set of models/stages of which SA is just one of several techniques, and thus quenching may be the final result.

Problems in mRNA splicing arise in determining alignments of large sequences, and here SQ was found to be a valuable tool [61].

A study on optimal selection of panels of rodent-human hybrid clones, involving sifting through as many as 10^{22} possible panels [62], found simulated annealing (no specific temperature schedule reported) better that random sampling or "random downhill" sampling (random sampling keeping the lowest cost function always, possibly getting trapped in a local optima).

A study on peptide conformations [63] used SQ on 102 parameters in a cost function determining the force fields. The constant c in the exponential temperature schedule,

$$T_{i+1} = c T_i \ , \tag{20}$$

was related to the variance of the free energy, itself related to the heat capacity. The use of this kind of complex cost function permitted the investigators to straightforwardly establish constraints on their cost function based directly on experimental and computational data.

The author has used SA (ASA) to fit a complex cost function modeling large-scale human neocortical activity to EEG (electroencephalographic) data [64]. Parameters in a model of columnar neuronal firing states (depicting 100's of neurons) were developed across multiple scales of interaction, into an electric potential model of larger scale brain activity (depicting millions of neurons), requiring a cost function expressed in terms of nonlinear Lagrangians, and as sets of discrete and continuous constraints.

Sets of complex constraints, i.e., sets of weights to suppress ambiguities arising from null space components that arise in singular value decompositions (SVD), were easily handled using an SQ algorithm to develop a cost function capable of fitting (and predicting) images in computer tomography (CT) scanning [65]. The authors were well aware of their use of SQ, explaining its appropriateness due to the simple surface and single minima of their cost function.

4.8. Physics

An early application of SA/SQ algorithms was to Ising models of magnetism, a prototypical system for problems in many disciplines. Spin-glass Ising systems present enormous numbers of local optima, and here it seems that only SA/SQ have a chance of finding a global optima. One such application used a new method of simulated tempering, outlined below, on the random field Ising model, a system presenting a very rough energy landscape [66].

The original Metropolis algorithm was designed for multivariate integrals of the kind that arise when performing path integrals [67]. SA has been used to enhance the efficiency of such calculations by first establishing extrema stationary phase points for these integrals [68], analogous to finding paths of maximum likelihood of integrals over multivariate dynamic probability distributions. Here, it is advantageous to keep track of multiple local minima as they arise, to include their contributions to the overall integral, and so in practice SQ is applied.

SA (ASA) can be substantially parallelized at several scales, and integrated with modern methods of functional stochastic calculus defining nonlinear Lagrangians, and with algorithms developed for a statistical mechanics of neocortical interactions, to produce a powerful generic mesoscopic neural network (MNN) [21,57]. This algorithm is quite generic, and can be used to process information in a wide class of systems, especially, but not limited to, those amenable to modeling by mathematical physics techniques alternatively described by path-integral Lagrangians, Fokker-Planck equations, or Langevin rate equations.

In the limit of quasi-linear Lagrangians (quadratic Lagrangian forms yielding quasi-linear stochastic differential equations) for which good mean-field limits exist, an SQ algorithm, mean-field annealing discussed below, can be more efficient and as effective as SA [30].

4.9. Geophysics

SQ techniques have proven useful in modeling seismic waveforms. It was noted that nonlinearity and stochasticity, which plague many other standard algorithms, can in fact be quite useful for such sparse data [69]. This made SA techniques extremely attractive and useful. Seismic traces over time and locations are “stacked” and the cost function to be minimized was taken to be the negative of the total stack power.

Another similar technique is to optimize a second order correlation function based on the products of offset stacks (the previous method is essentially with offset zero), which tends to minimize degeneracies associated with the previous method [70]. This was performed using an SQ algorithm.

Another paper reports the importance in applying an SQ algorithm of experimenting to find a good starting temperature near the “critical temperature” at which alternative optima are strongly searched [71].

4.10. Finance

Quite a few private communications with financial institutions have established that SA (or SQ?) is rapidly becoming an in-house algorithm of choice when dealing with financial instruments. Standard nested regression and local-search methods usually are applied to develop hybrid securities, e.g., combining markets in interest rates, foreign-exchange, equities, and commodities, by linking them via options, futures, forwards, and swaps, to increase profits and reduce risks in investments as well as in trading [72]. However, the complexity and nonlinearity of these multivariate systems, and the increasing interest of including information from more sophisticated modeling into trading rules, have called for more sophisticated numerical algorithms. For example, the author has written ASA codes to optimize trading rules, including optimization of sub-shells including models used to forecast variables, such as prices, volumes, open interests, momentums of these variables derived from Lagrangians representing multivariate nonlinear multiplicative-noise processes, etc. As such, the parameters are a mix of continuous and discrete sets, but these seem to be able to processed quite smoothly by ASA. One of the several strong features of these algorithms is their flexibility in accommodating many *ad hoc* constraints, rules, etc., as well as algebraic models. An example of the power of SA (ASA), coupled with new statistical mechanical modeling techniques, demonstrated that interest rates could be fit much better to data than previous published studies [73,74].

One study has used SQ on a set of several econometric problems [75], including cost functions arising in: the monetary theory of exchange rate determination, a study of firm production efficiency, and a neural net model which generates chaos reputed by some to mirror some economic and financial series. The authors demonstrated that their SQ algorithm performed better, e.g., at least more reliably finding more optima, than other numerical techniques such as a genetic algorithm and a quasi-Newton algorithm. In that study, using a Boltzmann algorithm [26], quenching is introduced by taking an exponential temperature schedule,

$$T_{i+1} = r_T T_i \ , \tag{21}$$

where r_T lies between 0 and 1, and is selected in an *ad hoc* way to aid the global and local aspects of the search. The problems they had in not always finding a global optima likely can be traced to using SQ, as well as to using the standard Boltzmann method which does not permit much room for addressing different sensitivities of the parameters, a point recognized by the authors. Their FORTRAN code is in the public domain [76].

4.11. Military

Optimal disbursement of resources is a common problem to large systems and is especially critical in defense. A study in optimal deployment of missile interceptors [77] used an SQ algorithm [78], permitting the acceptance criteria to get stricter as the temperature decreases, by multiplying the difference of saved and generated cost functions by the value of the saved cost function raised to an *ad hoc* power.

Tracking problems, in air, on the seas, and under water, present optimization problems to extract viable trajectories from sparse data. SQ techniques have been effective when used with detailed physical models that describe such trajectories [79].

A project to baseline computer wargames to exercise data required fitting data to multivariate nonlinear stochastic differential equations, expressed as an equivalent nonlinear Lagrangian [80]. SA (ASA) was used effectively to fit alternative models represented by such short-time conditional probabilities. Path-integral calculations then could be used to compare long-time correlations in these models with the data, to determine the best model.

5. Modifications/improvements on SA

5.1. SQ modifications

If one is willing to spend some time and resources learning a bit more about a system's cost function, and is willing to spend a bit more time experimenting with additional complexity, then likely one of several methods designed to enhance the efficiency of SQ can be chosen.

5.1.1. Acceleration/termination of annealing

Many modifications to SA are directly related to spending fewer resources addressing the perceived problem of too slow evolution of accepted configurations. This perception seems to be warranted in a large number of papers, some of which are referenced above, that have demonstrated they can achieve optimal solutions for their systems in less time than that mandated by the sufficiency conditions of the "proofs." Many approaches utilize the accepted to generated ratio to dynamically alter the annealing schedule, e.g., attempting to maintain a ratio ≈ 1 [26].

There has been much success in stopping SA rather arbitrarily at the later low-temperature stages of a search, when some low predetermined acceptance ratio is reached, then to proceed with another algorithm. A "rejectionless" method was developed, similar in spirit to this method, but more methodical, and yielding a search time not dependent on the acceptance ratio or temperature [81]. Acceptance criteria are biased according to information being gathered on the cost function during the search, maintaining detailed balance throughout the search. They suggest using standard SA until some low acceptance criteria is reached, then to finish annealing using their method.

Some theoretical work that has been performed, enhancing the efficiency of generating functions, utilizes techniques borrowed from stochastic relaxation [82], biasing the acceptance criteria using information gathered during the search coupled with the Metropolis algorithm [44]. This inhomogeneous (nonstationary) method works particularly well when the cost function can be approximately represented by the log of an additive Gaussian distribution typical of problems arising in image-restoration. While aware of the schedule required for SA using their Metropolis-based algorithm, their numerical work utilized a much faster geometric SQ schedule,

$$T_k = T_0 r^k \ . \tag{22}$$

In many complex problems, much CPU time is spent inefficiently until an annealing temperature is reached that starts to effectively sample the local minima. One solution to this is to perform some short sample runs [83], perhaps using as much as 80% of the total CPU time (saving the best-to-date optima), to determine this effective temperature.

A study in mapping tasks onto network graphs examined some correlations between graph regularity and annealing schedules using SA and SQ algorithms [84]. Their conclusions lead them to the hypothesis that further research into ultrametricity [85], a theory including the topology and regularity of multiple scales, may lead to more efficient SA algorithms and to guidelines when to expect SQ to perform well on a given class of problems.

5.1.2. Using knowledge of mathematical/physical structure

An approach to minimizing time spent in unproductive regions, the essence of “importance sampling” introduced by the Monte Carlo approach [10], is to combine the SA algorithm with the “hybrid Monte Carlo” method [86], which has been implemented in a neural network study [87]. The idea of the hybrid Monte Carlo method is to expand the “configuration space” (q) of an energy-type cost function $E(q)$, creating a Hamiltonian-type cost function $H(p, q)$ defined in “phase space” (p, q),

$$H(p, q) = E(q) + \tfrac{1}{2} p^2 \ , \tag{23}$$

where p is the “momentum.” In addition to stochastic movements typical of SA and Monte Carlo methods, “dynamic” moves are also made in $H \approx$ constant surfaces, permitting some rejection of high H values. A few other interesting twists are added, e.g., defining the annealing temperature $T \propto p^2$, permitting additional control at the various stages of the search process [87].

This is an SQ algorithm, but one which permits a better physical understanding of the physical processes of the state space in a given problem than by merely accelerating the temperature schedule. For example, the schedule for the stochastic momentum vector updates was selected to be

$$p_{t+1} = \alpha |p_t| u + (1 - \alpha^2)^{1/2} n \ , \ 0 \le \alpha < 1 \ , \tag{24}$$

where u is a random vector of unit length with uniformly distributed direction, and n is a multivariate Gaussian random vector. It can be demonstrated that this procedure is equivalent to sampling the phase space according to the Boltzmann distribution for this system.

Simply relating the temperature to the square of the momentum in phase space, after a Fourier transform of the phase space cost function H with respect to p in $(p \cdot q)$-space, is equivalent to considering the configuration cost function as having a variance proportional to the temperature, i.e., the form of the standard algorithm. For physical systems driven by differential rate equations with simple white (additive Gaussian Markovian) noise η, this is equivalent to a Langevin equation

$$\frac{dq}{dt} = -\nabla E(q) + (2T)^{1/2} \eta \ , \tag{25}$$

which prompted a hybrid annealing/Langevin algorithm [3]. This permits the treatment of nonstationary systems.

When the system is nonlinear, multivariate, and the noise contains multiplicative nonconstant functions, a Riemannian geometry is induced [88], which requires some care in the definition of the cost function used for SA. Several problems have been successfully treated with ASA using the equivalent nonlinear Lagrangian specified for such systems, to define a maximum likelihood fitting procedure [16,57,64,73,80].

Care must be taken in minimizing the cost functions of many physical systems by terminating the temperature schedule [66]. At finite temperature, the optimal equilibrium state of such a system is determined by the free energy, which usually is much more difficult to calculate than just the internal energy.

5.1.3. Mean-field annealing (MFA)

An SQ algorithm gaining in popularity that quenches in a different manner than manipulating the annealing schedule, but often which is combined with other SQ techniques such as described previously, is mean-field annealing (MFA).

The value and justification for MFA is derived from energy/cost functions quadratic in the variables being quenched, e.g., similar to the Ising model of magnetic interactions among spins [52],

$$H(s) = \sum_i s_i (h_i + 2 \sum_{j \ne i} V_{ij}) + \sum_{k \ne i} h_k + \sum_{k \ne i} \sum_{j \ne k,i} V_{kj} s_k s_j \ , \tag{26}$$

where the (discrete) vector s_i is the variable being quenched, h_i is the "external field," and V_{ij} is the pair interaction. It is noted that the mean value $< s_i >$, $< \cdot >$ taken with respect the the Boltzmann distribution in terms of the effective field Φ_i (the coefficient of the factor multiplying s_i in the first term on the RHS of the above equation), relaxes to equilibrium much faster than does the stochastic variable s_i. The algorithm proceeds by performing updates at each decreasing temperature level: calculating the present $< s_i >$, then updating Φ_i, and then updating $< s_i >$.

For quasi-quadratic energy functions, the mean values of the variables is a good approximation to the optimal stochastic state. Then, the MFA fit efficiently capitalizes on searching for these deterministic most likely trajectories, instead of having to perform a fully stochastic search. I.e., when a mean-field theory is a good approximate to a stochastic cost function, then MFA may be a very efficient alternative to SA.

This algorithm has drawn further interest because such quasi-quadratic energy functions, defined in terms of path integrals over their Boltzmann probability distributions, possess variational equations which can directly lead to mean-field equations for their variables. For example, this is similar to deriving the force law F for a spring,

$$F = -k < x >= ma = m < \ddot{x} > \ , \tag{27}$$

using the variational principle on the distribution of the Lagrangian L,

$$L = \frac{1}{2} m \dot{x}^2 - \frac{1}{2} k x^2 \ , \tag{28}$$

corresponding to the Hamiltonian H,

$$H = \frac{1}{2} p^2/m + \frac{1}{2} k x^2 \ . \tag{29}$$

The interest arises from taking $< s_j >$ as a sigmoidal function and making various identifications between all parameters, arriving at a set of differential equations describing a neural network as well as the TSP discussed above. Furthermore, these sets of equations can be simply mapped into software and hardware parallel architectures [30,53,54]. Of course, the Ising model is a prototype for many other systems, but here the attraction is to use a neural net to perform the search.

While the analogies and applications using MFA are interesting, most results are for (quasi-)linear systems, albeit high dimensional ones. It should be kept in mind that these methods likely work as well as other SQ methods, and likely fail as well on more nonlinear problems which require more exhaustive searches.

5.1.4. Optimal ensembles

Ensemble algorithms have been developed to approach the selection of optimal schedules. The basic approach is to generate independent Markov chains in the annealing algorithm, i.e., using different initial conditions/random seeds for each trajectory. This technique was parallelized and applied with very good results to TSP problems, adding some adaptive quenching designed to take advantage of specific system features [89]. These features include *a priori* selecting aspects of the associated distribution to be optimized, e.g., its mean, median, mode, energy cut-offs, etc. They also include some adaptive features based on information being gathered during the fit, such as coarse graining the current optima to an equivalent tree structure over the mountain-like terrain of the cost function, using information being gathered by all generated points, to estimate the current relaxation time to equilibrium at a given temperature. These estimates are used to estimate an optimal ensemble size.

5.1.5. Simulated tempering (ST)

A method of simulated tempering (ST) has been designed for maximizing probabilities, $P(X, m)$, with cost functions $H(X)$, by expanding the variable space, X, adding a new discrete variable, m [66],

$$P(X, m) \propto \exp\bigl(\beta_m H(X) - g_m\bigr) , \tag{30}$$

where β_m are considered to be dynamic temperatures, and g_m are *a priori* assigned constants. This approach selects Δg_m to approximate an intermediate value of $\beta_m H(X)$ as the index m changes, but such that ΔH is appreciable with contiguous changes in β_m in order to obtain an efficient algorithm. An important feature is to maintain equilibrium of the system while seeking alternative minima and lowering the effective cost function according to the m schedule. Care must be taken not to select the m schedule inappropriately or else premature quenching can result.

5.1.6. Prejudicial search

Several of the above algorithms blend some form of deterministic search together with SA to achieve efficiency beyond that of SA alone, albeit at the risk of not guaranteeing convergence to the global optima within finite times. A formal argument has been presented to demonstrate that a class of such “prejudicial” searches can achieve asymptotic convergence to the optimum state, albeit this can sacrifice statistical convergence in finite time, and this typically does require some *a priori* knowledge of the system [90].

5.2. Ergodic SA improvements

5.2.1. Fast annealing (FA)

Although there are many variants and improvements made on the “standard” Boltzmann algorithm described above, many textbooks finish just about at this point without going into more detail about other algorithms that depart from this explicit algorithm [4]. Specifically, it was noted that the Cauchy distribution has some definite advantages over the Boltzmann form [13]. The Cauchy distribution,

$$g(\Delta x) = \frac{T}{(\Delta x^2 + T^2)^{(D+1)/2}} , \tag{31}$$

has a “fatter” tail than the Gaussian form of the Boltzmann distribution, permitting easier access to test local minima in the search for the desired global minimum.

It is instructive to note the similar corresponding heuristic demonstration, that the Cauchy $g(\Delta x)$ statistically finds a global minimum. If Eq. (8) is replaced by

$$T(k) = \frac{T_0}{k} , \tag{32}$$

then here

$$\sum_{k_0}^{\infty} g_k \approx \frac{T_0}{\Delta x^{D+1}} \sum_{k_0}^{\infty} \frac{1}{k} = \infty . \tag{33}$$

Note that the “normalization” of g has introduced the annealing-time index k, giving some insights into how to construct other annealing distributions. The method of FA is thus seen to have an annealing schedule exponentially faster than the method of BA. This method has been tested in a variety of problems [13].

5.2.2. Adaptive simulated annealing (ASA)

In a variety of physical problems we have a D-dimensional parameter-space. Different parameters have different finite ranges, fixed by physical considerations, and different annealing-time-dependent sensitivities, measured by the derivatives of the cost-function at local minima. BA and FA have distributions which sample infinite ranges, and there is no provision for considering differences in each parameter-dimension; e.g., different sensitivities might require different annealing schedules. This prompted the development of a new probability distribution to accommodate these desired features [16], leading to a variant of SA that in fact justifies an exponential temperature annealing schedule. This algorithm is discussed in more detail in the next section.

5.2.3. Parallel annealing

Advantage can be taken of parallel processing to implement SA at several stages, e.g., during random number generation as well as in calculating generating functions [57]. However, such approaches likely do not maximally take advantage of the possibilities of parallel processing.

5.2.3.1. Mob parallel annealing

Efficiency in implementing standard SA was achieved by adding a “mob” heuristic in graph-embedding problems, to randomly swap coarse-grained neighborhoods of parameter space, thereby avoiding spending large amounts of resources in local minima [91]. In their work, a mob is a collection of vertices that have a “large” effect on the cost function when swapped. Another advantage of this heuristic is that it readily permits parallel implementation. The authors prove that standard SA techniques become innately serial at low temperatures, preventing strict parallelism at massive scales. The proof makes some assumptions about the probabilities of rejection at low temperatures that should be examined for other SA algorithms. In practice, they find ample parallelism for many of their problems.

As mentioned in the section above on SQ modifications, in the context of ultrametricity, the general concept of respecting multiple scales of the cost function may be one of the most important areas for future investigation in improving the performance of SA.

5.2.3.2. Time-homogeneous parallel annealing

There are more sophisticated algorithms that have proven that parallel probabilistic exchanges of information gathered from processors annealing at different temperatures can increase the overall rate of convergence [92]. The proof given depends on selecting the final and initial temperatures, and all intermediate cycles, before the search begins. It is instructive here to examine one of the bases on which they develop the bias factors for their algorithm. They define $p(T, E, T', E',)$ as the probability of exchange between two solutions (to be used in periodically exchanging information between processors), with the logical requirement that solutions are always exchanged if a better one is found at a higher temperature,

$$\Delta T \, \Delta E = (T - T')(E - E') < 0 \Rightarrow p = 1 \ .$$

For the other cases, they invoke detailed balance,

$$\frac{1}{Z(T)} \exp(-\frac{E}{T}) \frac{1}{Z(T')} \exp(-\frac{E'}{T'}) p(T, E, T', E') = \frac{1}{Z(T)} \exp(-\frac{E'}{T}) \frac{1}{Z(T')} \exp(-\frac{E}{T'}) \ , \tag{34}$$

$$Z(T) = \sum_x \exp(-\frac{E(x)}{T}) \ . \tag{35}$$

This yields

$$p(T, E, T', E') = \begin{cases} 1 & \text{if } \Delta T \, \Delta E < 0 \\ \exp(-\frac{\Delta T \, \Delta E}{TT'}) & \text{otherwise .} \end{cases} \tag{36}$$

They note that this enables p to be calculated without knowledge of the partition function Z, making updates much simpler than other techniques requiring the calculation of Z.

5.2.3.3. Parallel recombinative simulated annealing (PRSA)

A hybrid algorithm of parallel recombinative simulated annealing (PRSA), blending desirable features of genetic algorithms (GA), briefly discussed above, with standard SA has been proposed [24,93]. Crossover and mutation techniques of GA are performed during various stages of SA. Parallelism is a feature typically incorporated with GA, and this is added here as well. If two conditions are strictly satisfied—(a) that the system can move to an optimal solution in a finite number of transitions, and (b) there exists detailed balance symmetry condition—then the global convergence properties of SA are retained. This method requires generation of points until two trial cost functions are found that both differ from each other and from the latest saved cost function by some threshold value θ. This paper also contains an short review of some other work in parallel simulated annealing.

5.2.3.4. Parallel systolic SA

The use of arrays of transputers, each independent computing element containing a complete independent database with periodic communication between elements, presents opportunities for parallel SA [94]. Especially at low temperatures with higher rejection rates, independent Markov chains can be efficient in exploring large spaces. This study also used FA, described above, which presented approximately constant rates of acceptances at low temperatures, relatively independent of the temperature, permitting efficient optimizations of crystallization problems.

5.2.3.5. Analysis of Gibbs sampler

A contribution to rigorously investigating SA, and the possibilities to be gained by parallel processing, is some work done examining rates of convergence at any constant temperature applied to lattice problems [95]. They investigate conditions under which the equilibrium Boltzmann distribution is achieved for some simple models, which possess this distribution as a limit when sampled sequentially, when parallelized.

5.2.4. Global versus local stages of search

There are some aspects of SA that can be modified to improve efficiency without sacrificing the sufficiency conditions. The initial stages of SA generally are committed to global diffuse searching, to explore alternative optima. Here, it is important to examine sensitivity of the global search to initial temperature. The final stages of SA generally are committed to local focused convergence into the global optima. If some information is known about the system, then a natural criteria might be established.

In dealing with the overall process, it may help to appreciate just how the products of generating and acceptance probabilities describe a Markov chain. Many practitioners experiment with their particular application to implement the temperature schedule. For example, some practitioners establish sets of homogeneous (time or temperature independent) Markov chains at constant temperature, each set of length determined until a predetermined number of repetitive visits to a given optima occur. This can serve to permit more global searching in particularly rough parameter landscapes, reducing long local visits. For example, in a study utilizing FA [51], better results were obtained using such sets of homogeneous Markov chains.

In applications using ASA [21,64,73,80], typically using a low acceptance to generated ratio, long local visits were reduced by changing the acceptance distribution temperature only on each new acceptance. This seems to work well since only the generating distributions are fat-tailed and can more reasonably follow the fastest temperature schedule permitted by the sufficiency conditions.

Furthermore, in problems where the desired degree of accuracy and precision are known beforehand, it may be reasonable to shunt over to a more efficient local algorithm. In some ASA applications, good results have been obtained shunting over to the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [96] after a determined number of acceptances did not improve the precision of the minimum by a predetermined amount.

6. Adaptive simulated annealing (ASA)

In a variety of physical problems we have a D-dimensional parameter-space. Different parameters have different finite ranges, fixed by physical considerations, and different annealing-time-dependent sensitivities, measured by the curvature of the cost-function at local minima. BA and FA have g distributions which sample infinite ranges, and there is no provision for considering differences in each parameter-dimension, e.g., different sensitivities might require different annealing schedules. These are among several considerations that gave rise to Adaptive Simulated Annealing (ASA). Full details are available by obtaining the publicly available source code [6].

ASA considers a parameter α^i_k in dimension i generated at annealing-time k with the range

$$\alpha^i_k \in [A_i, B_i] \ , \tag{37}$$

calculated with the random variable y^i,

$$\alpha^i_{k+1} = \alpha^i_k + y^i(B_i - A_i) \ ,$$

$$y^i \in [-1, 1] \ . \tag{38}$$

Define the generating function

$$g_T(y) = \prod_{i=1}^{D} \frac{1}{2(|y^i| + T_i)\ln(1 + 1/T_i)} \equiv \prod_{i=1}^{D} g^i_T(y^i) \ . \tag{39}$$

Its cumulative probability distribution is

$$G_T(y) = \int_{-1}^{y^1} \cdots \int_{-1}^{y^D} dy'^1 \cdots dy'^D \, g_T(y') \equiv \prod_{i=1}^{D} G^i_T(y^i) \ ,$$

$$G^i_T(y^i) = \frac{1}{2} + \frac{\mathrm{sgn}\,(y^i)}{2} \frac{\ln(1 + |y^i|/T_i)}{\ln(1 + 1/T_i)} \ . \tag{40}$$

y^i is generated from a u^i from the uniform distribution

$$u^i \in U[0, 1] \ ,$$

$$y^i = \mathrm{sgn}\,(u^i - \frac{1}{2}) T_i [(1 + 1/T_i)^{|2u^i - 1|} - 1] \ . \tag{41}$$

It is straightforward to calculate that for an annealing schedule for T_i

$$T_i(k) = T_{0i} \exp(-c_i k^{1/D}) \ , \tag{42}$$

a global minima statistically can be obtained. I.e.,

$$\sum_{k_0}^{\infty} g_k \approx \sum_{k_0}^{\infty} [\prod_{i=1}^{D} \frac{1}{2|y^i| c_i}] \frac{1}{k} = \infty \ . \tag{43}$$

It seems sensible to choose control over c_i, such that

$$T_{fi} = T_{0i} \exp(-m_i) \text{ when } k_f = \exp n_i \ ,$$

$$c_i = m_i \exp(-n_i/D) \ , \tag{44}$$

where m_i and n_i can be considered “free” parameters to help tune ASA for specific problems.

It has proven fruitful to use the same type of annealing schedule for the acceptance function h as used for the generating function g, i.e., Equations (42) and (44), but with the number of acceptance points, instead of the number of generated points, used to determine the k for the acceptance temperature.

New parameters α^i_{k+1} are generated from old parameters α^i_k from

$$\alpha^i_{k+1} = \alpha^i_k + y^i (B_i - A_i) \ , \tag{45}$$

constrained by

$$\alpha^i_{k+1} \in [A_i, B_i] \ . \tag{46}$$

I.e., y^i's are generated until a set of D are obtained satisfying these constraints.

6.1. Reannealing

Whenever doing a multi-dimensional search in the course of a real-world nonlinear physical problem, inevitably one must deal with different changing sensitivities of the α^i in the search. At any given annealing-time, it seems sensible to attempt to “stretch out” the range over which the relatively insensitive parameters are being searched, relative to the ranges of the more sensitive parameters.

It has proven fruitful to accomplish this by periodically rescaling the annealing-time k, essentially reannealing, every hundred or so acceptance-events, in terms of the sensitivities s_i calculated at the most current minimum value of the cost function, $\underline{L}$,

$$s_i = \partial \underline{L} / \partial \alpha^i \ . \tag{47}$$

In terms of the largest $s_i = s_{\max}$, it has proven fruitful to reanneal by using a rescaling for each k_i of each parameter dimension,

$$k_i \rightarrow k'_i \ ,$$

$$T'_{ik'} = T_{ik} (s_{\max} / s_i) \ ,$$

$$k'_i = \big(\ln(T_{i0} / T_{ik'}) / c_i\big)^D \ . \tag{48}$$

T_{i0} is set to unity to begin the search, which is ample to span each parameter dimension.

The acceptance temperature is similarly rescaled. In addition, since the initial acceptance temperature is set equal to a trial value of $\underline{L}$, this is typically very large relative to the global minimum. Therefore, when this rescaling is performed, the initial acceptance temperature is reset to the most current minimum of $\underline{L}$, and the annealing-time associated with this temperature is set to give a new temperature equal to the lowest value of the cost-function encountered to annealing-date.

Also generated are the “standard deviations” of the theoretical forms, calculated as $[\partial^2 \underline{L} / (\partial \alpha^i)^2]^{-1/2}$, for each parameter α_i. This gives an estimate of the “noise” that accompanies fits to stochastic data or functions. At the end of the run, the off-diagonal elements of the “covariance matrix” are calculated for all parameters. This inverse curvature of the theoretical cost function can provide a quantitative assessment of the relative sensitivity of parameters to statistical errors in fits to stochastic systems.

A few other twists can be added, and such searches undoubtedly will never be strictly by rote. Physical systems are so different, some experience with each one is required to develop a truly efficient algorithm.

6.2. Self optimization

Another feature of ASA is its ability to recursively self optimize its own Program Options, e.g., the c_i parameters described above, for a given system. An application is described below.

6.2.1. Quenching

Another adaptive feature of ASA is its ability to perform quenching. This is applied by noting that the temperature schedule above can be redefined as

$$T_i(k_i) = T_{0i} \exp(-c_i k_i^{Q_i / D}) \ ,$$

$$c_i = m_i \exp(-n_i Q_i / D) \ , \tag{49}$$

in terms of the “quenching factor” Q_i. The above proof fails if $Q_i > 1$ as

$$\sum_k \prod^D 1/k^{Q_i/D} = \sum_k 1/k^{Q_i} < \infty \; . \qquad (50)$$

This simple calculation shows how the "curse of dimensionality" arises, and also gives a possible way of living with this disease. In ASA, the influence of large dimensions becomes clearly focussed on the exponential of the power of k being $1/D$, as the annealing required to properly sample the space becomes prohibitively slow. So, if we cannot commit resources to properly sample the space ergodically, then for some systems perhaps the next best procedure would be to turn on quenching, whereby Q_i can become on the order of the size of number of dimensions.

The scale of the power of $1/D$ temperature schedule used for the acceptance function can be altered in a similar fashion. However, this does not affect the annealing proof of ASA, and so this may used without damaging the (weak) ergodicity property.

6.3. ASA applications

The above defines this method of adaptive simulated annealing (ASA), previously called very fast simulated reannealing (VFSR) [16] only named such to contrast it the previous method of fast annealing (FA) [13]. The annealing schedules for the temperatures T_i decrease exponentially in annealing-time k, i.e., $T_i = T_{i0} \exp(-c_i k^{1/D})$. Of course, the fatter the tail of the generating function, the smaller the ratio of acceptance to generated points in the fit. However, in practice, it is found that for a given generating function, this ratio is approximately constant as the fit finds a global minimum. Therefore, for a large parameter space, the efficiency of the fit is determined by the annealing schedule of the generating function.

A major difference between ASA and BA algorithms is that the ergodic sampling takes place in an $n + 1$ dimensional space, i.e., in terms of n parameters and the cost function. In ASA the exponential annealing schedules permit resources to be spent adaptively on reannealing and on pacing the convergence in all dimensions, ensuring ample global searching in the first phases of search and ample quick convergence in the final phases. The acceptance function $h(\Delta x)$ chosen is the usual Boltzmann form satisfying detailed balance, and the acceptance-temperature reannealing paces the convergence of the cost function to permit ergodic searching in the n-parameter space considered as the independent variables of the dependent cost function.

ASA has been applied to several systems, ranging from combat analysis [80,97], to finance [73,74], to neuroscience [64], to a set of test problems [21], to a new technique combining the power of SA with the physics of large-scale systems [57], to many other systems [98].

ASA source code in C-language is publicly available [6]. It is certain that there is much research to be done on determining optimal or even reasonable ASA parameters, for different classes of systems, especially in higher dimensional spaces of user parameters. A major purpose of making this code publicly available is to motivate more of this research, and thus make the code more useful to a wider audience.

6.4. ASA annealing versus quenching

As an example of applying some of the features of ASA, the reannealing, self optimization, and quenching features were applied to the difficult test problem in the code, given above as Eq. (15), for dimensions $n = 4$ and $n = 8$, containing 10^{5n} minima. Relative to previously published ASA/VFSR studies that were faster and more accurate than other global optimization algorithms, these options can speed up the search (number of cost_function calls) by as much as a factor of 20, without losing accuracy in finding the global minimum.

For dimension 4, quenching values of Q were chosen as 1, 2, 3, and 4, and "super-quenching" values of 8 and 16 were also run. For dimension 8, quenching values of Q were chosen as 1, 2, 3, 4, 5, 6, 7, and 8, and "super-quenching" values of 16 and 24 were also run. Separate runs were done for cases with reannealing and with no reannealing.

It was quite surprising to see that all runs achieved the global optima of 0. It should be noted that the super-quenched runs quickly exhausted the preset limits of precision of 10^{-30}, which became the values of the parameter and cost function annealing temperatures rather soon into into the runs. The runs continued, which simply translated into no annealing after those low temperatures were reached, just randomly searching within the narrowed ranges. If the limits of precision were set lower, then it should be expected that the runs would end much faster, albeit they might not attain the global minimum. This likely explains why the super-quenching did not perform much better than for values of $Q = n$.

For each value of Q selected, three trajectories were run. Figs. 4 and 5 give results for $n = 4$, for the cases of no reannealing and with reannealing, respectively. Figs. 6 and 7 give results for $n = 12$, for the cases of no reannealing and with reannealing, respectively. Tables I and II give values of the final number of calls generated for each trajectory, for dimensions $n = 4$ and $n = 8$, respectively.

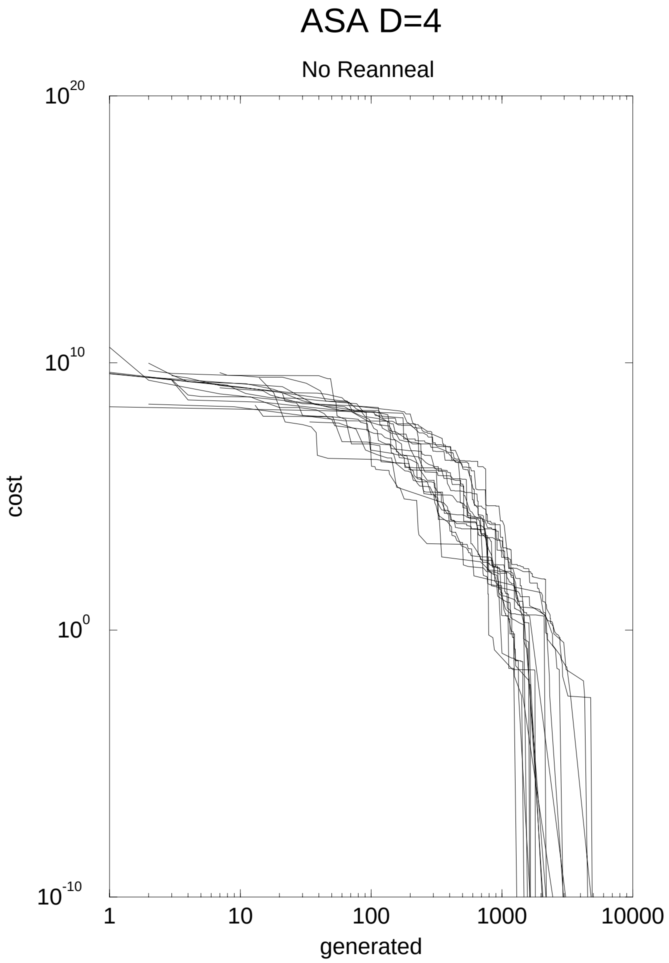

Fig. 4. Superimposed are runs for $n = 4$, the case of no reannealing, 3 trajectories each for cases of $Q = 1$, 2, 3, 4, 8, and 16. Although the actual final cost function values are 0, they were set to 10^{-10} for purposes of this log-log plot. See Table I for assignments of trajectories with their final numbers of required calls.

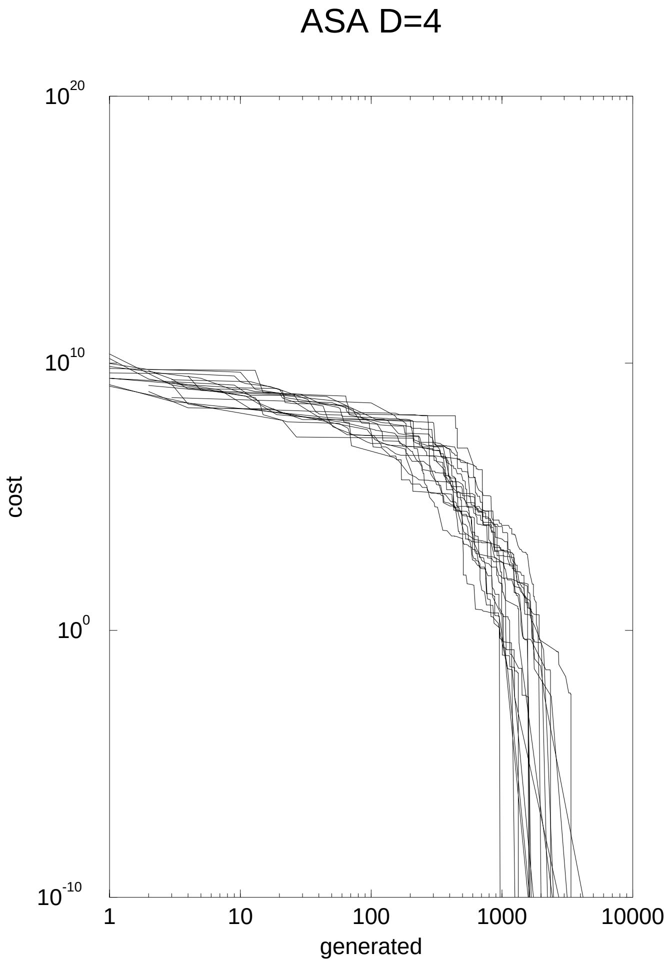

Fig. 5. Superimposed are runs for $n = 4$, the case including reannealing, 3 trajectories each for cases of $Q = 1, 2, 3, 4, 8$, and 16. Although the actual final cost function values are 0, they were set to 10^{-10} for purposes of this log-log plot. See Table I for assignments of trajectories with their final numbers of required calls.

Dimension = 4

	No_Reanneal		Reanneal	
Quench	Traj	Calls	Traj	Calls
1	a	4814	d	3163
1	b	4523	e	3375
1	c	4910	f	4189
2	a	1473	d	1340
2	b	2065	e	1635
2	c	3078	f	1749
3	a	1637	d	2728
3	b	1807	e	1614
3	c	1300	f	968
4	a	2938	d	1259
4	b	2953	e	2230
4	c	1570	f	1632
8	a	1647	d	1682
8	b	2447	e	2355
8	c	2188	f	1999
16	a	1660	d	1593
16	b	2035	e	2440
16	c	2171	f	2486

Table I. The values of the number of generated calls to achieve the global optimum point are given for dimensions $n = 4$, for several values of quenching values of Q, for three trajectories each for cases of no reannealing (trajectories a, b and c) and with reannealing (trajectories d, e and f).

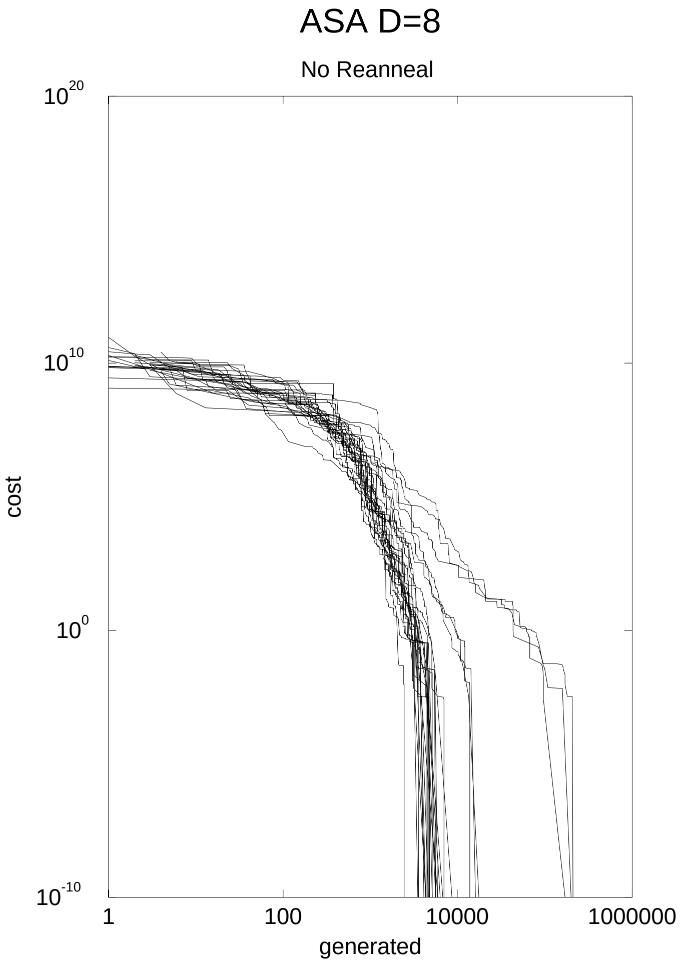

Fig. 6. Superimposed are runs for $n = 8$, the case of no reannealing, 3 trajectories each for cases of $Q = 1$, 2, 3, 4, 5, 6, 7, 8, 16, and 24. Although the actual final cost function values are 0, they were set to 10^{-10} for purposes of this log-log plot. See Table II for assignments of trajectories with their final numbers of required calls.

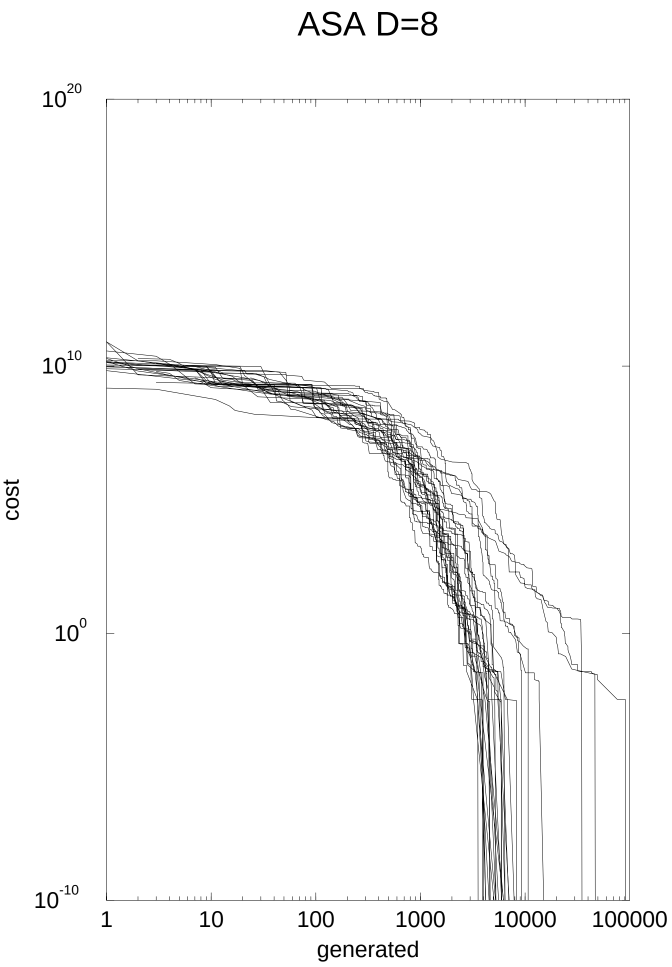

Fig. 7. Superimposed are runs for $n = 8$, the case including reannealing, 3 trajectories each for cases of $Q = 1, 2, 3, 4, 5, 6, 7, 8, 16$, and 24. Although the actual final cost function values are 0, they were set to 10^{-10} for purposes of this log-log plot. See Table II for assignments of trajectories with their final numbers of required calls.

Dimension = 8

	No_Reanneal		Reanneal	
Quench	Traj	Calls	Traj	Calls
1	a	211066	d	46850
1	b	170163	e	35121
1	c	199969	f	91993
2	a	16115	d	10763
2	b	13800	e	15080
2	c	17554	f	9340
3	a	7026	d	8291
3	b	8661	e	6439
3	c	5613	f	5275
4	a	5668	d	5289
4	b	3497	e	3920
4	c	4388	f	4557
5	a	4637	d	6222
5	b	2458	e	7006
5	c	4327	f	7873
6	a	5975	d	6428
6	b	6328	e	4207
6	c	3585	f	4590
7	a	4347	d	5255
7	b	4692	e	6246
7	c	4807	f	5992
8	a	5920	d	7051
8	b	4142	e	4014
8	c	5128	f	5976
16	a	5640	d	4663
16	b	4534	e	3573
16	c	4355	f	5955
24	a	5147	d	5048
24	b	6776	e	4206
24	c	4807	f	5607

Table II. The values of the number of generated calls to achieve the global optimum point are given for dimensions $n = 8$, for several values of quenching values of Q, for three trajectories each for cases of no reannealing (trajectories a, b and c) and with reannealing (trajectories d, e and f).

7. Conclusion

Simulated annealing is a very powerful and important tool in a variety of disciplines. However, often it is not applied according to strict adherence to sufficiency conditions permitting the researcher to truly claim that the optimal solution has been (statistically) found. The reason typically given is simply that many variants of this technique are considered to be too consuming of resources to be applied in such strict fashion.

There exist faster variants of true simulated annealing (SA), but these apparently are not as quite easily coded and so they are not widely used. More well-documented user-friendly code, e.g., menu-driven, would definitely help.

Many modifications of SA are really quenching, and should aptly be called simulated quenching (SQ). However, SQ is not without its usefulness and often is justified in terms of its practicality. Many current research problems, given the complexity of models and constraints faced, likely would not be approached if there was not some chance of reasonable solution of the optimization problems that arise. SQ does fill this void. Explicit calculations on a difficult test problem have demonstrated that at least in some cases SQ can perform faster than SA with the same accuracy.

The best solution to this situation might occur if further research could establish some necessary conditions on SA for obtaining an optimal solution with fewer resources than the current sufficiency conditions permit. The search for such a “holy grail” is not quite hopeless. As we have tried to outline here, so many difficult problems have been treated quite favorably by SQ techniques, that it seems that a classification of problems might exist, yielding insight into what “shade” of SA or SQ might be most appropriate for a given system.

Acknowledgements

Many of the authors cited here generously responded to my electronic mail requests for (p)reprints on current work in this field; quite a few read earlier drafts and contributed their feedback. Their timely response and helpful suggestions are gratefully acknowledged. Graphs were produced using XVGR (graphics for exploratory data analysis), a public domain software package running under UNIX and X11, developed by Paul Turner at the Oregon Graduate Institute.

References

1. S.-K. Ma, *Statistical Mechanics*, World Scientific, Philadelphia, (1985).
2. D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon, Optimization by simulated annealing: An experimental evaluation (Parts 1 and 2), Report, AT&T Bell Laboratories, Murray Hill, NJ, (1987).
3. S.B. Gelfand, Analysis of simulated annealing type algorithms, Ph.D. Thesis, MIT, Cambridge, MA, (1987).
4. P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel, Dordrecht, The Netherlands, (1987).
5. N.E. Collins, R.W. Egelese, and B.L. Golden, Simulated annealing — an annotated bibliography, *Am. J. Math. Management Sci.* **8** (3 & 4), 209-307 (1988).
6. L. Ingber, Adaptive Simulated Annealing (ASA), [ftp.alumni.caltech.edu: /pub/ingber/ASA-shar, ASA-shar.Z, ASA.tar.Z, ASA.tar.gz, ASA.zip], Lester Ingber Research, McLean, VA, (1993).
7. A. Charnes and M. Wolfe, Extended Pincus theorems and convergence of simulated annealing, *Int. J. Systems Sci.* **20** (8), 1521-1533 (1989).
8. M. Pincus, A Monte Carlo method for the approximate solution of certain types of constrained optimization problems, *Oper. Res.* **18**, 1225-1228 (1970).
9. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, Optimization by simulated annealing, *Science* **220** (4598), 671-680 (1983).
10. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21** (6), 1087-1092 (1953).
11. S. Geman and D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images, *IEEE Trans. Patt. Anal. Mac. Int.* **6** (6), 721-741 (1984).
12. V. Cerny, A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, Report, Comenius University, Bratislava, Czechoslovakia, (1982).
13. H. Szu and R. Hartley, Fast simulated annealing, *Phys. Lett. A* **122** (3-4), 157-162 (1987).
14. K. Binder and D. Stauffer, A simple introduction to Monte Carlo simulations and some specialized topics, in *Applications of the Monte Carlo Method in Statistical Physics*, (Edited by K. Binder), pp. 1-36, Springer-Verlag, Berlin, (1985).
15. J. Mathews and R.L. Walker, *Mathematical Methods of Physics, 2nd ed.*, Benjamin, New York, NY, (1970).
16. L. Ingber, Very fast simulated re-annealing, *Mathl. Comput. Modelling* **12** (8), 967-973 (1989).
17. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, (1975).
18. K.A. De Jong, Genetic algorithms are NOT function optimizers, in *Foundations of Genetic Algorithms: Proceedings 24-29 July 1992*, (Edited by D. Whitley), Morgan Kaufman, Vail, CO, (1992).
19. Z. Michalewicz and C.Z. Janikow, Genetic algorithms for numerical optimization, *Statistics Computing* **1**, 75-91 (1991).
20. J.D. Schaffer, L.J. Eshelman, and D. Offutt, Spurious correlations and premature convergence in genetic algorithms, in *Foundations of Genetic Algorithms*, (Edited by G. Rawlins), pp. 102-112, Morgan Kaufmann, San Mateo, CA, (1991).
21. L. Ingber and B. Rosen, Genetic algorithms and very fast simulated reannealing: A comparison, *Mathl. Comput. Modelling* **16** (11), 87-100 (1992).
22. G.E. Liepins, On fixed points and global convergence of genetic algorithms, Report, Oak Ridge National Laboratory, Oak Ridge, TN, (1992).
23. D.B. Fogel and J.W. Atmar, Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems, *Biol. Cybern.* **63**, 111-114 (1990).

24. D.E. Goldberg, A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing, *Complex Sys.* **4**, 445-460 (1990).

25. M.K. Sen and P.L. Stoffa, Comparative analysis of simulated annealing and genetic algorithms: Theoretical aspects and asymptotic convergence, *Geophys.* , (submitted) (1993).

26. A. Corana, M. Marchesi, C. Martini, and S. Ridella, Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm, *ACM Trans. Mathl. Software* **13** (3), 262-279 (1987).

27. L. Ingber and B. Rosen, Very Fast Simulated Reannealing (VFSR), [ringer.cs.utsa.edu: /pub/rosen/vfsr.Z], University of Texas, San Antonio, TX, (1992).

28. N.N. Schraudolph and J.J. Grefenstette, A Users Guide to GAUCSD 1.2, Report, University of California at San Diego, La Jolla, CA, (1991).

29. B. Rosen, Function optimization based on advanced simulated annealing, *IEEE Workshop on Physics and Computation - PhysComp '92* , 289-293 (1992).

30. P.D. Simic, Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations, *Network* **1**, 89-103 (1990).

31. E. Bonomi and J.-L. Lutton, The *N*-city travelling salesman problem: Statistical mechanics and the metropolis algorithm, *SIAM Rev.* **26** (4), 551-568 (1984).

32. Q. Wu and T.H. Sloane, CMOS leaf-cell design using simulated annealing, Report, Bucknell University, Lewisburg, PA, (1992).

33. M. Murray, J.B. Burr, D.S. Stork, M.-T. Leung, K. Boonyanit, G.J. Wolff, and A.M. Peterson, Deterministic Boltzmann machine VLSI can be scales using multi-chip modules, in *Application Specific Array Processors*, (Edited by J. Fortes, E. Lee and T. Meng), pp. 206-217, IEEE Computer Society Press, Los Alamitos, CA, (1992).

34. R.S. Bucy and R.S. DiEsposti, Decision tree design by simulated annealing, *Math. Modelling Numer. Anal.* , (submitted) (1992).

35. D.K. Garnick, Y.H.H. Kwong, and F. Lazebnik, Algorithmic search for extremal graphs of girth at least five, Report, Bowdoin College, Brunswick, ME, (1992).

36. A. Jagota, Efficiently approximating *Mac-Clique* in a Hopfield-style network, in *International Joint Conference on Neural Networks*, pp. 248-353, IEEE, Baltimore, MD, (1992).

37. J.R. Elliot and P.B. Gibbons, The construction of subsquare free Latin squares by simulated annealing, *Australasian J. Combinatorics* **5**, 209-228 (1992).

38. C. Peterson and J.R. Anderson, Neural networks and NP-complete optimization problems: A performance study on the graph bisection problem, *Complex Sys.* **2**, 59-89 (1988).

39. W.N. Venables and J.A. Eccleston, Randomized search strategies for finding near optimal block and row-column designs, Report, University of Adelaide, Australia, (1992).

40. C.V. Packer, Applying row-column permutation to matrix representations of large citation networks, *Infor. Processing Management* **25** (3), 307-314 (1989).

41. R.W. Klein and R.C. Dubes, Experiments in projection and clustering by simulated annealing, *Pattern Recognition* **22** (2), 213-220 (1989).

42. J. Kelly, B. Golden, and A. Assad, Using simulated annealing to solve controlled rounding problems, *ORSA J. Comput.* **2** (2), 174-185 (1990).

43. H. Raittinen and K. Kaski, Image deconvolution with simulated annealing method, *Physica Scripta* **T33**, 126-130 (1990).

44. G. Bilbro, Efficient generators in simulated annealing, Report TR-91/12, North Carolina State University, Raleigh, NC, (1991).

45. M.J. Black and P. Anandan, Robust dynamic motion estimation over time, in *Proc. Comput. Vis. Patt. Recog., CVPR-91, Maui, HI, 3-6 June 1991*, pp. 296-302, IEEE Computer Society Press, Los Alamitos, CA, (1991).

46. K. Kishida, Physical Langevin model and the time-series model in systems far from equilibrium, *Phys. Rev. A* **25**, 496-507 (1982).

47. K. Kishida, Equivalent random force and time-series model in systems far from equilibrium, *J. Math. Phys.* **25**, 1308-1313 (1984).

48. L. Ingber, Statistical mechanics algorithm for response to targets (SMART), in *Workshop on Uncertainty and Probability in Artificial Intelligence: UC Los Angeles, 14-16 August 1985*, (Edited by P. Cheeseman), pp. 258-264, American Association for Artificial Intelligence, Menlo Park, CA, (1985).

49. D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, A learning algorithm for Boltzmann machines, *Cog. Sci.* **9**, 147-169 (1985).

50. J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, (1991).

51. P. Israel and C. Koutsougeras, An annealing approach to associative recall in the CBM model, Report, Tulane University, New Orleans, LA, (1992).

52. G. Bilbro, R. Mann, T.K. Miller, W.E. Snyder, D.E. Van der Bout, and M. White, Optimization by mean field annealing, in *Advances in Neural Network Information Processing Systems*, (Edited by D. Touretzky), pp. 91-98, Morgan-Kaufman, San Mateo, CA, (1989).

53. P.D. Simic, Constrained nets for graph matching and other quadratic assignment problems, *Neural Comput.* **3** (2)(1991).

54. C. Peterson and B. Söderberg, Artificial neural networks and combinatorial optimization problems, in *Local Search in Combinatorial Optimization*, (Edited by E.H.L. Aarts and J.K. Lenstra), pp. (to be published), John Wiley & Sons, New York, NY, (1992).

55. J. Niittylahti, Hardware implementation of Boolean neural network using simulated annealing, Report 8-92, Tampere University of Technology, Tampere, Finland, (1992).

56. J.M. Goodwin, B.E. Rosen, and J.J. Vidal, Image recognition and reconstruction using associative magnetic processing, *Int. J. Patt. Recog. Artif. Intell.* **6** (1), 157-177 (1992).

57. L. Ingber, Generic mesoscopic neural networks based on statistical mechanics of neocortical interactions, *Phys. Rev. A* **45** (4), R2183-R2186 (1992).

58. H. Bohr and S. Brunak, A travelling salesman approach to protein conformation, *Complex Sys.* **3**, 9-28 (1989).

59. D. Bouzida, S. Kumar, and R.H. Swendsen, A simulated annealing approach for probing biomolecular structures, in *Proceedings of Hawaii International Conference on Systems Sciences (HICSS)*, pp. (to be published), Carnegie Mellon University, Pittsburgh, PA, (1993).

60. D.L. Dowe, J. Oliver, T.I. Dix, L. Allison, and C.S. Wallace, A decision graph explanation of protein secondary structure prediction, in *Proceedings of 26th Hawaii International Conference on Systems Sciences (HICSS)*, pp. (to be published), Carnegie Mellon University, Pittsburgh, PA, (1993).

61. A.V. Lukashin, J. Engelbrecht, and S. Brunak, Multiple alignment using simulated annealing: branch point definition in human mRNA splicing, *Nucleic Acids Res.* **20** (10), 2511-2516 (1992).

62. T.M. Goradia and J. Lange, Applications of coding theory to the design of somatic cell hybrid panels, *Math. Biosci.* **91**, 201-219 (1988).

63. D.E. Palmer, C. Pattaroni, K. Nunami, R.K. Chadha, M. Goodman, T. Wakamiya, K. Fukase, S. Horimoto, M. Kitazawa, H. Fujita, A. Kubo, and T. Shiba, Effects of dehydroalanine on peptide conformations, *J. Am. Chem. Soc.* **114** (14), 5634-5642 (1992).

64. L. Ingber, Statistical mechanics of neocortical interactions: A scaling paradigm applied to electroencephalography, *Phys. Rev. A* **44** (6), 4017-4060 (1991).

65. H. Haneishi, T. Masuda, N. Ohyama, T. Honda, and J. Tsujiuchi, Analysis of the cost function used in simulated annealing for CT image reconstruction, *Appl. Optics* **29** (2), 259-265 (1990).

66. E. Marinari and G. Parisi, Simulated tempering: A new Monte Carlo scheme, *Europhys. Lett.* , (submitted) (1992).

67. L.S. Schulman, *Techniques and Applications of Path Integration*, J. Wiley & Sons, New York, (1981).

68. T.L. Beck, J.D. Doll, and D.L. Freeman, Locating stationary paths in functional integrals: An optimization method utilizing the stationary phase Monte Carlo sampling function, *J. Chem. Phys.* **90** (6), 3181-3191 (1989).

69. D.H. Rothman, Nonlinear inversion, statistical mechanics, and residual statics estimation, *Geophys.* **50**, 2784-2796 (1985).

70. K. Vasudevan, W.G. Wilson, and W.G. Laidlaw, Simulated annealing statics computation using an order-based energy function, *Geophys.* **56**, 1831-1839 (1991).

71. M.K. Sen and P.L. Stoffa, Nonlinear one-dimensional seismic waveform inversion using simulated annealing, *Geophys.* **56**, 1624-1638 (1991).

72. J.F. Marshall and V.K. Bansal, *Financial Engineering: A Complete Guide to Financial Innovation*, New York Institute of Finance, New York, NY, (1992).

73. L. Ingber, Statistical mechanical aids to calculating term structure models, *Phys. Rev. A* **42** (12), 7057-7064 (1990).

74. L. Ingber, M.F. Wehner, G.M. Jabbour, and T.M. Barnhill, Application of statistical mechanics methodology to term-structure bond-pricing models, *Mathl. Comput. Modelling* **15** (11), 77-98 (1991).

75. W.L. Goffe, G.D. Ferrier, and J. Rogers, Global optimization of statistical functions with simulated annealing, *J. Econometrics* **60** (1/2), 65-100 (1994).

76. W.L. Goffe, G.D. Ferrier, and J. Rogers, Simulated annealing, [research.att.com: /netlib/opt/simann.f.Z], AT&T Bell Labs, Murray Hill, NJ, (1992).

77. I.O. Bohachevsky, M.E. Johnson, and M.L. Stein, Optimal deployment of missile interceptors, *Am. J. Math. Management Sci.* **8** (3 & 4), 361-387 (1988).

78. I.O. Bohachevsky, M.E. Johnson, and M.L. Stein, Generalized simulated annealing for function optimization, *Technometrics* **28** (3), 209-217 (1986).

79. W.A. Kuperman, M.D. Collins, J.S. Perkins, and N.R. Davis, Optimal time-domain beamforming with simulated annealing including application of *a priori* information, *J. Acoust. Soc. Am.* **88** (4), 1802-1810 (1990).

80. L. Ingber, H. Fujio, and M.F. Wehner, Mathematical comparison of combat computer models to exercise data, *Mathl. Comput. Modelling* **15** (1), 65-90 (1991).

81. J.A. Greene and K.J. Supowit, Simulated annealing without rejected moves, *IEEE Trans. Comput.-Aided Design* **CAD-5** (1), 221-228 (1986).

82. W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57** (1), 97-109 (1970).

83. A. Basu and N. Frazer, Rapid determination of the critical temperature in simulated annealing inversion, *Science* **249**, 1409-1412 (1990).

84. C. Lee and L. Bic, The effect of structure in the mapping problem using simulated annealing, Report, University of California, Irvine, (1991).

85. R. Rammal, G. Toulouse, and M.A. Virasoro, Ultrametricity for physicists, *Rev. Mod. Phys.* **58** (3), 765-788 (1986).

86. S. Duane, A.D. Kennedy, B.J. Pendelton, and D. Roweth, Hybrid Monte Carlo, *Phys. Lett. B* **195**, 216-222 (1987).

87. R.M. Neal, Bayesian training of backpropagation networks by the hybrid Monte Carlo method, Report CRG-TR-92-1, University of Toronto, Toronto, Canada, (1992).

88. F. Langouche, D. Roekaerts, and E. Tirapegui, *Functional Integration and Semiclassical Expansions*, Reidel, Dordrecht, The Netherlands, (1982).

89. K.H. Hoffmann, D. Würtz, C. de Groot, and M. Hanf, Concepts in optimizing simulated annealing schedules: an adaptive approach for parallel and vector machines, in *Lecture Notes in Economics and Mathematical Systems 367*, (Edited by M. Grauer and D.B. Pressmar), pp. 154-174, Springer Verlag, Heidelberg, (1991).

90. N.E. Cotter, J.B. Soller, and T.J. Guillerm, Optimization by prejudicial search, Report, University of Utah, Salt Lake City, UT, (1992).

91. J.E. Savage and M.G. Wloka, Parallelism in graph-partitioning, *J. Parallel Distrib. Comput.* **13**, 257-272 (1991).

92. K. Kimura and K. Taki, Time-homogeneous parallel annealing algorithm, Report TR-673, Institute for New Generation Computer Technology, Tokyo, Japan, (1991).

93. S.W. Mahfoud and D.E. Goldberg, Parallel recombinative simulated annealing: A genetic algorithm, IlliGAL Report No. 92002, University of Illinois, Urbana, IL, (1992).

94. A. ter Laak, L.O. Hertzberger, and P.M.A. Sloot, Nonconvex continuous optimization experiments on a transputer system, in *Transputer Systems — Ongoing Research*, (Edited by A. Allen), pp. 251-265, IOS Press, Amsterdam, Holland, (1992).

95. P.A. Ferrari, A. Frigessi, and R. Schonmann, Convergence of some partially parallel Gibbs sampler with annealing, *Ann. Appl. Probab.* , (to be published) (1993).

96. D.F. Shanno and K.H. Phua, Minimization of unconstrained multivariate functions, *ACM Trans. Mathl. Software* **2**, 87-94 (1976).

97. L. Ingber and D.D. Sworder, Statistical mechanics of combat with human factors, *Mathl. Comput. Modelling* **15** (11), 99-127 (1991).

98. M. Wofsey, Technology: Shortcut tests validity of complicated formulas, *The Wall Street Journal* **222** (60), B1 (1993).