JOURNAL OF COMPUTATIONAL PHYSICS 56, 259–271 (1984)

A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables

DAVID VANDERBILT AND STEVEN G. LOUIE

Department of Physics, University of California at Berkeley, Berkeley, California 94720

Received November 1, 1983

Numerical optimization methods based on thermodynamic concepts are extended to the case of continuous multidimensional parameter spaces. Techniques which allow this strategy to be implemented efficiently and reliably, including a self-regulatory mechanism for choosing the random step distribution, are described. The method is applied to a set of standard global minimization problems, and to a typical non-linear least-squares functional fitting problem.

I. INTRODUCTION

Recently, Kirkpatrick *et al.* [1] (KGV) have pointed out a fruitful connection between statistical mechanics and problems of combinatorial optimization. Physical systems, they noted, may be coaxed into a minimum energy configuration, such as that of a crystal, by a slow annealing process. The reduction of the temperature confines the system to a smaller and smaller region of phase space, but is carried out slowly enough to allow the system to pass out of metastable local energy minima. In this way, the system arrives at the global minimum configuration, which may occupy a minute fraction of the original phase space. Just as Metropolis Monte Carlo computer calculations can be used to simulate such a physical minimization process [2], they pointed out, so can they be used to solve arbitrary numerical optimization problems in which one searches for a global minimum of some function (analogous to energy) defined over some multivariate parameter space (analogous to physical degrees of freedom). This simulated annealing approach to optimization has already proven to be a powerful numerical tool [1] and is an elegant example of the ability of physical concepts to inform other fields of science.

KGV have applied this approach to a variety of combinatorial problems, such as the traveling salesman problem and computer circuit design problems. In these cases, the free parameters take on discrete values; the "steps" of the Monte Carlo random walk correspond to permutations in the list of cities to be visited, interchanges of circuit elements, or other discrete operations. While such combinatorial optimization problems are certainly of wide applicability, there is also a large variety of problems which involve minimization with respect to continuous parameters. For example, if one desires to fit a function to a sum of exponentials (or some other parametrized functions), the choice of the decay constants is an optimization problem. Continuous

0021-9991/84 $3.00

global optimization problems also frequently arise in many other contexts, including engineering design, econometrics, data analysis, etc. Traditional local minimization methods, such as Newton–Raphson or quasi-Newton local descent approaches [3], can suffer severely from trapping in local minima for such problems.

The principal complication introduced in going from the discrete to the continuous application of Monte Carlo simulated annealing is that the choice of the random steps becomes more subtle. In general, the optimal magnitude and directions of the vector steps are not known in advance. Steps which are too small will be very inefficient in exploring the phase space; steps which are too large will always be rejected. If the function minimum lies in a highly anisotropic valley, steps which explore in directions perpendicular to the valley axis are generally wasted. Finally, the size of the steps should shrink as the temperature is reduced and the volume of accessible phase space shrinks. The principal innovation introduced here is a self-regulatory mechanism for the step distribution which automatically insures that an efficient choice of step size and anisotropy is maintained throughout the anneal.

The plan of the paper is as follows. In Section II we review the Metropolos Monte Carlo algorithm, show how it can be applied to continuous optimization, and describe the proposed step distribution self-regulation mechanism. Application is made in Section III to a family of standard continuous global optimization problems. Because the performance of the best previously available global optimization algorithms is already tabulated in the literature for this set of problems [4], we can make a direct comparison and show that the present method is quite comparable to these in efficiency. In Section IV we illustrate how the Monte Carlo annealing approach can be used to solve non-linear least-squares fitting problems. The particular problem illustrated here is the fitting of a radial function to a sum of Gaussians. This problem arose in the context of constructing potentials for electronic band structure calculations, but the approach is equally applicable to any functional fitting problem. Finally, Section V contains a summary and conclusions.

II. Method

Consider a function $E(x_1, x_2, ..., x_n) = E(\mathbf{x})$ defined over an n-dimensional continuous parameter space. The problem is to minimize E with respect to $\mathbf{x}$, where E may be the energy of a physical system, the error in a fitting problem, or any other "objective function." The Metropolis Monte Carlo algorithm [2] proceeds by choosing an initial starting point $\mathbf{x}_0$ and making random steps $\Delta\mathbf{x}$. At each step, the change

$$\Delta E = E(\mathbf{x} + \Delta\mathbf{x}) - E(\mathbf{x}) \tag{1}$$

in the objective function is evaluated. If ΔE is negative, the step is accepted. If ΔE is positive, the step is accepted with a probability

$$p = \exp[-\Delta E/T]. \tag{2}$$

The series of accepted steps then generates a random walk which explores the parameter space, and which at long times is governed by the probability distribution function

$$P(\mathbf{x}) = \frac{1}{Z} \exp[-E(\mathbf{x})/T] \tag{3}$$

where $P(\mathbf{x})\, d^n x$ is the probability that the walk will be in the volume $d^n x$ on any given step at long times, and the normalization constant or "partition function" Z is given by

$$Z = \int d^n x \exp[-E(\mathbf{x})/T]. \tag{4}$$

The parameter T plays the role of temperature; as T is decreased slowly, the volume $\Omega(T)$ of phase space with non-negligible $P(\mathbf{x})$ shrinks until the system is eventually forced to "freeze" or "anneal" into the configuration of lowest E. If the anneal is carried out slowly enough, the system will avoid getting trapped in local minima, because Eq. (2) does allow steps which increase E temporarily to get over a barrier into a new local (or global) minimum.

A natural way to choose the random steps is to call a random number generator n times to generate the numbers $(u_1, u_2, \ldots, u_n)$ where each u_i is chosen independently from the interval $[-\sqrt{3}, \sqrt{3}]$ (i.e., with zero mean and unit variance). The resulting vector $\mathbf{u}$ occurs with a probability density $g(\mathbf{u})$ which is constant inside a hypercube of volume $(2\sqrt{3})^n$ and zero outside. We then choose the step $\Delta\mathbf{x}$ according to

$$\Delta\mathbf{x} = \mathbf{Q} \cdot \mathbf{u} \tag{5}$$

where the matrix $\mathbf{Q}$ controls the step distribution. The simplest choice would be $Q_{ij} = a\delta_{ij}$ which generates an isotropic distribution with RMS average step length $a\sqrt{n}$. In general we can characterize the step distribution due to $\mathbf{Q}$ by the covariance matrix

$$\begin{aligned} s_{ij} &= \int d^n u\, g(\mathbf{u})\, \Delta x_i\, \Delta x_j \\ &= (2\sqrt{3})^{-n} \int_{-\sqrt{3}}^{\sqrt{3}} du_1 \cdots du_n \left(\sum_k Q_{ik} u_k\right)\left(\sum_l Q_{jl} u_l\right) \\ &= \sum_k Q_{ik} Q_{jk}. \end{aligned} \tag{6}$$

Inverting the procedure, we can generate random steps with any desired covariance matrix $\mathbf{s}$ by solving

$$\mathbf{s} = \mathbf{Q} \cdot \mathbf{Q}^T \tag{7}$$

for $\mathbf{Q}$, e.g., via Choleski decomposition, and then using Eq. (5) to generate the $\Delta\mathbf{x}$'s.

In order to implement the algorithm, we need a recipe for reducing T, and for modifying the covariance matrix $\mathbf{s}$, as a functin of time into the anneal. The efficiency of the algorithm will depend strongly on how this is done.

We use a simple schedule for reducing T: carry out M steps at T_0, then M steps at $\chi_T T_0$, then M more at $\chi_T^2 T_0$, etc. The initial temperature may be set equal to the variance of random sample of $E(\mathbf{x})$'s if no other information is available. The geometric temperature reduction factor, $0 < \chi_T < 1$, should be chosen by trial and error; χ_T too small will cause trapping in local minima, while χ_T too large will unnecessarily waste computer time.

The algorithm for choosing the covariance matrix $\mathbf{s}$ is more complicated. It should be chosen so that on a given set of M steps, the random walk more or less explores the entire extent of the available phase space $\Omega(T)$, which we may rather fuzzily define to be

$$\Omega(T) = \{\mathbf{x} \mid E(\mathbf{x}) - E_{\min}(\mathbf{x}) \lesssim T\}. \tag{8}$$

If the steps are too small, almost all steps will be accepted; if too large, almost all will be rejected. In either case, there is very little information gained per step. The optimal efficiency will be obtained with the optimal rate of information gain, i.e., when approximately half of the steps are accepted. Moreover, the "shape" of $\mathbf{s}$ is important; if we think of the tensor $\mathbf{s}$ as describing an ellipsoid, the ellipsoid should more or less match the shape of $\Omega(T)$. For example, in the quadratic region of a minimum, we should have $\mathbf{s}$ proportional to the inverse of the Hessian,

$$\mathbf{s}^{-1} \propto \left. \frac{\partial^2 E}{\partial x_i \, \partial x_j} \right|_{\mathbf{x}_{\min}}. \tag{9}$$

If the axes of $\mathbf{s}$ are poorly aligned with the topography of $E(\mathbf{x})$, much time will be wasted exploring fruitless directions of search in phase space.

The method we propose is based on using the excursions of the random walk itself as a measure of the local topography. At the end of the lth set of M steps, we calculate the first and second moments of the walk segment:

$$A_i^{(l)} = \frac{1}{M} \sum_{m=1}^{M} x_i^{(m;\,l)} \tag{10}$$

$$S_{ij}^{(l)} = \frac{1}{M} \sum_{m=1}^{M} [x_i^{(m;l)} - A_i^{(l)}][x_j^{(m;l)} - A_j^{(l)}] \tag{11}$$

where $\mathbf{x}^{(m;l)}$ is the value of $\mathbf{x}$ on the mth step of the lth set. Unlike $\mathbf{s}$, which describes the probability distribution of the individual trial steps, $\mathbf{S}$ describes the shape of an actual segment of the walk. For a free random walk, i.e., one in which no steps are rejected, we would expect on average that

$$\langle \mathbf{S}_{\text{free}}^{(l)} \rangle = \beta M \mathbf{s}^{(l)}. \tag{12}$$

To be precise, the brackets indicate an average over the nM random variables $\Delta x_i^{(m;l)}$ which generate this segment of the walk. A straightforward calculation gives $\beta = \frac{1}{6}$ for the arithmetic average. We then propose to choose $\mathbf{s}$ for the next iteration according to the prescription

$$\mathbf{s}^{(l+1)} = \frac{\chi_s}{\beta M} \mathbf{S}^{(l)} \tag{13}$$

so that

$$\langle \mathbf{S}_{\text{free}}^{(l+1)} \rangle = \chi_s \mathbf{S}^{(l)}. \tag{14}$$

The "growth factor" χ_s is chosen > 1 (typically $\chi_s = 3$) so that a free random walk on the $(l+1)$th set would cover, on average, $\chi_s^{1/2}$ times as much ground in each direction as on the previous iteration. In fact we have used a β based on a geometric average, since the size of the walk after many free sets is really a product, rather than a sum, of the random growth factors. A one-dimensional numerical calculation gives $\beta = 0.11$; this value has been adopted. To review the algorithm, then, the step size distribution is recalculated at the end of each set of M steps using Eqs. (10), (11), and (13), and is implemented on the next set using Eqs. (7) and (5).

The result of this procedure is illustrated schematically in Fig. 1. Suppose that the initial $\mathbf{s}$ is too small, so that at first all steps are small and almost all are accepted. Then Eqs. (12) and (14) apply and the size of the region covered by the walk grows by $\chi_s^{1/2}$ on each iteration. Eventually the walk grows to the size of $\Omega(T)$ and starts hitting the walls imposed by $E(\mathbf{x})$ ($l = 6$ in Fig. 1); at this point a large number of steps begin to be rejected. Now $\mathbf{S}$, and therefore $\mathbf{s}$, begin to reflect the shape of $\Omega(T)$ and Eq. (12) no longer holds. As T is reduced, the walk is constrained to $\Omega(T)$ as it

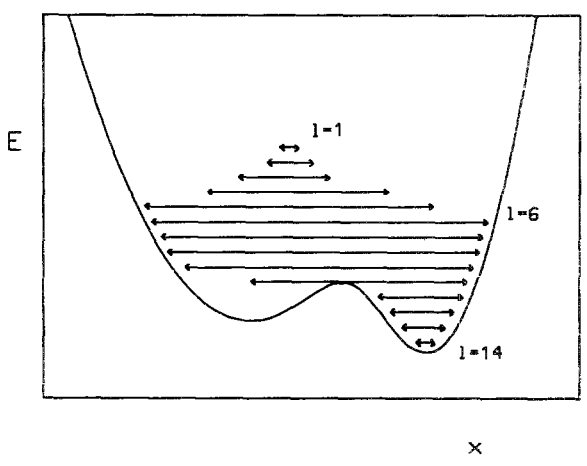

FIG. 1. Schematic illustration of random walk distribution during an anneal. Arrows indicate extent of random walk of the lth set of steps.

shrinks, and **s** is automatically maintained at a size appropriate to $\Omega(T)$. Moreover, if $\Omega(T)$ is highly anisotropic, the random walks will be forced (by rejected steps) to have large excursions in the free directions and will be highly constrained in other directions; this will reflect itself in **S** and hence in **s**, so that on subsequent sets the search will be primarily along the free directions. In other words, **s** automatically adapts itself to the local topography of $E(\mathbf{x})$.

In practice, M should be chosen large enough so that the random walk on each set is long enough to generate reasonable statistics for Eq. (11). At a minimum we require $M > n$; otherwise **S** is singular. In practice, we choose $M \approx 15n$; empirically this works well for small n, e.g., $n \lesssim 8$. For large-dimensional systems, larger values of M appear to be needed to get good statistics for the shape of S. If $\mathbf{s} \propto \mathbf{I}$ in Eq. (12) we expect on average $\mathbf{S}_{\text{free}} \propto \mathbf{I}$, but as **S** has n eigenvalues the chances are that one or more will be anomalously large or small for large n. The small eigenvalues are troublesome, because the progress of the random walk is arrested in the direction corresponding to the eigenvector. Several approaches to this problem present themselves. One could choose

$$\mathbf{s}^{(l+1)} = \alpha \frac{\chi_s}{\beta M} \mathbf{S}^{(l)} + (1-\alpha)\,\mathbf{s}^{(l)} \tag{15a}$$

where the damping constant $0 < \alpha < 1$ controls the rate at which information from **S** is folded into **s**. If the objective function is known to be fairly well behaved, one could require **s** to remain isotropic,

$$\mathbf{s}^{(l+1)} = \frac{\chi_s}{\beta M} [\det\ \mathbf{S}^{(l)}]^{1/n} \mathbf{I} \tag{15b}$$

or, more generally, aligned with the cartesian axes:

$$\mathbf{s}_{ij}^{(l+1)} = \frac{\chi_s}{\beta M} S_{ii}^{(l)} \delta_{ij}. \tag{15c}$$

The brute force solution is just to increase M. We have not carried out careful tests to determine which strategy is the most generally effective.

The uniform annealing schedule embodied by a constant χ_T may not be the most efficient procedure; often there is a crucial temperature regime in which slow annealing will help to choose between local minima. KGV suggest that the "specific heat"

$$C(T) = \frac{d}{dt} \langle E(T) \rangle \tag{16}$$

(where the brackets indicate a themal avarage, i.e., an average over steps of the walk), or equivalently

$$C(T) = \frac{1}{T^2} [\langle E^2 \rangle - \langle E \rangle^2] \tag{17}$$

should be used as an indicator of such a temperature regime [1]. By the statistical mechanics analogy, a peak in $C(T)$ indicates that the configuration is becoming ordered, i.e., frozen into a minimum. Slower annealing in this temperature regime can be used to insure that the walk is not trapped in this minimum unless it is the global minimum. Note that for reasonable statistics, Eq. (17), and not the finite difference approximation to Eq. (16), should be used.

Finally, one needs to know when to stop the algorithm. For this purpose we compute $\langle E \rangle$ and $E_{\min}$ for each set of M steps, and stop if

$$\frac{\langle E \rangle - E_{\min}}{\langle E \rangle} < \eta. \tag{18}$$

For typical applications we use $\eta = 10^{-3}$. We find that the difference between $E_{\min}$ and the true local minimum is generally much smaller than this for small-dimensional problems, i.e., η is a conservative error estimate. Some efficiency could perhaps be gained by stopping earlier and going over to a local minimizer such as a quasi-Newton [3] or simplex [5] algorithm, but we have not done so.

III. Application to Standard Optimization Problems

A survey of known global optimization routines and their efficiency as measured on a standard set of seven global optimization problems has recently been published [4]. This survey represents the results of several years of international cooperation by a variety of research groups, oriented towards identifying the best algorithms available. For the sake of comparison, we have applied our method to the same set of problems, and used the same measures of efficiency in analyzing the results. The seven functions and our choice of search parameters for them are indicated in Table I. The precise definition of each function, and the volume of parameter space to be searched (in all cases a finite hypercube), may be found in Ref. [4].

In all cases we use $M = 15n$, $\chi_s = 3$, $\eta = 10^{-3}$. T_0 and χ_T are the principal free parameters; T_0 was essentially chosen on the basis of the variance of a random sample of $E(\mathbf{x})$'s, and χ_T was chosen by trial and error. In the case of F1, F2, and F3, the slow anneal at $\chi_T = 0.99$ was allowed to give way to a faster anneal at $\chi_T = 0.4$ when $\eta = 10^{-1}$ was reached, in order to avoid a needlessly slow local optimization within the domain of the minimum.

Table II gives the results compared with other methods. The algorithm was independently repeated 100 times for each function, and the average number of function evaluations and average computer time (in units of 1000 evaluations of function F1) are reported. Also shown is the probability of arrival at the global minimum using the present approach, based upon the sample of 100 runs. In all cases a local minimum was found. Even in the worst case, F1, the global minimum could be found with high certainty by running the program several times and choosing the

TABLE I

Seven Standard Functions and Search Parameters Used in Their Minimization

Function	n	Functional form[a]	M	T_0	χ_T	χ_s	η
F1	4	$-\sum_{i=1}^{5}[(\mathbf{x}-\mathbf{a}_i)^2+c_i]^{-1}$	60	0.5	0.99[b]	3.0	10^{-3}
F2	4	$-\sum_{i=1}^{7}[(\mathbf{x}-\mathbf{a}_i)^2+c_i]^{-1}$	60	0.5	0.99[b]	3.0	10^{-3}
F3	4	$-\sum_{i=1}^{10}[(\mathbf{x}-\mathbf{a}_i)^2+c_i]^{-1}$	60	0.5	0.99[b]	3.0	10^{-3}
F4	3	$-\sum_{i=1}^{4} c_i \exp[-\mathbf{x}^T\cdot\mathbf{A}_i\cdot\mathbf{x}]$	45	1.0	0.6	3.0	10^{-3}
F5	6	$-\sum_{i=1}^{4} c_i \exp[-\mathbf{x}^T\cdot\mathbf{A}_i\cdot\mathbf{x}]$	90	1.0	0.4	3.0	10^{-3}
F6	2	$a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos x_1+e$	30	20.0	0.1	3.0	10^{-3}
F7	2	8th-order polynomial in (x_1, x_2)	30	100.0	0.4	3.0	10^{-3}

[a] For the values of the parameters in the functions see Ref. [4].
[b] χ_T was reduced at the later stages of the anneal; see text.

TABLE II

Comparison of Efficiency of Global Minimization Algorithms, Giving the Number of Function Evaluations and Computer Time (in Units of 1000 F1 Evaluations), Respectively

Function	Search cluster[a]	Search cluster[b]	Controlled random search[c]	Bayesian[d]	P function[e]	Present approach	P_{global} (%)
F1	3679/10	7085/19	3800/14	1174/—	620/23	3910/16	54
F2	3606/13	6684/23	4900/20	1279/—	788/20	3421/15	64
F3	3874/15	7352/23	4400/20	1209/—	1160/30	3078/15	81
F4	2584/8	6766/17	2400/8	513/—	732/16	1224/4	100
F5	3447/16	11125/48	7600/46	1232/—	807/21	1914/12	62
F6	2499/4	1495/2	2500/3	362/—	378/15	557/1	100
F7	1558/4	1318/3	1800/4	189/—	597/14	1186/2	99

Note. The last column is the arrival probability for the global minimum using the present approach.
[a] Ref. [6].
[b] Ref. [7].
[c] Ref. [8].
[d] Ref. [9]; computer times are unavailable for this case.
[e] Ref. [10].

best minimum to arise. In general we find that the Monte Carlo annealing approach ranks fairly high among the known algorithms. The overhead costs (i.e., those unrelated to function evaluations) are low compared to other methods.

It is interesting to note which functions give the most difficulty for the annealing approach. The functions F1–F3 are difficult because the local minima are deep, highly isolated, and their depths are not apparent except in a small volume near the core of each minimum. The values of c_i in Table I for the lowest minima are about 0.2, while the search space is $x_j \in [0, 10]$, a volume $\Omega = 10^4$ [4]. In the vicinity of a minimum we can approximate

$$E \approx -\frac{1}{c} + \frac{1}{c^2}|\mathbf{x} - \mathbf{a}_i|^2 \tag{19}$$

and the accessible volume is about T^2c^4, so that a rough model for the partition function is

$$Z = T^2c^4 \exp[1/cT] + \Omega. \tag{20}$$

At temperatures T less than a critical temperature T_c, energy considerations win and the "particle" sits in the minimum [i.e., the first term of Eq. (20) dominates], while at $T > T_c$ entropy wins and the particle wanders on the plateau. Equating the two terms in Eq. (20) gives $T_c \approx 0.4$ for the lowest minima. The sudden transition from the plateau to the minimum is reflected in a sharp cusp in $C(T)$; Eq. (16) gives $C(T_c) \approx 100$ and width $\Delta T \approx 0.03$ fo this cusp. For $T < T_c$ the time constant for hopping between minima, $\tau \propto \exp[1/cT]$, is so long that the hopping probability is negligible. Thus the algorithm does sometimes get trapped in local minima, as shown in Table II. The algorithm actually works better for the "more complicated" functions F2 and F3, presumably because the barriers between minima are reduced.

In the case of the function F5, the problem is a competition between the global and lowest local minima, which are separated by ΔE of only ~ 0.12 [4]. Moreover, the global minimum is much the narrower of the two (by a factor of ~ 16 in volume) so that not until $T < 0.12/\ln(16) \approx 0.04$ will the global minimum be favored. At this temperature the hopping rate is again almost negligible, and trapping in the local minimum can occur.

The difference between an "easy" and a "difficult" function for the simulated annealing approach is illustrated in Fig. 2. The specific heat was calculated from Eq. (17) for F1 and F4 by annealing over a wide temperature range using $\chi_T = 0.8$. and the results averaged over 20 scans. (This may involve more work than needed for a single minimization, but it can be done once for a single representative of a class of problems.) We always find $C \to 0$ as $T \to \infty$ and $C \approx n/2$ as $T \to 0$ as expected for a hard-walled box and a quadratic minimum, respectively. We see signs of the sharp cusps in the specific heat for F1, indicating that very slow annealing is necessary in the interval $0.1 < T < 0.5$. For F4, on the other hand, we see only a gentle bump, and we can anneal through it quickly.

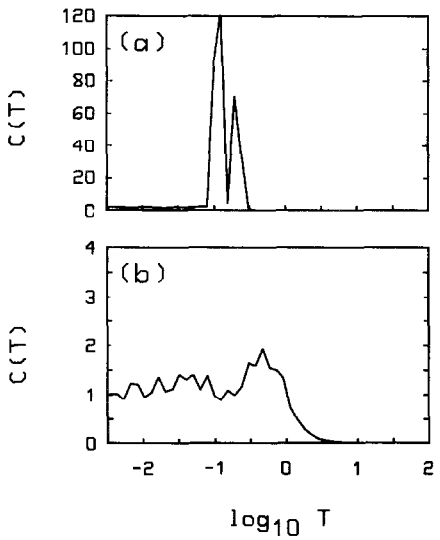

FIG. 2. Specific heat as a function of temperature as determined from an average over 20 anneals for (a) function F1 and (b) function F4. Note the difference in the vertical scales.

In summary, we find that the Monte Carlo simulated annealing approach performs well for this class of multiextremal problems. In most cases we would suggest running the anneal more than once to reduce the chances of missing the global minimum; this reduces the efficiency somewhat. On the other hand, use of a better local optimizer in the later stages of the procedure could improve the performance. Perhaps the present approach could be integrated with other approaches, such as search clustering [6] or the simplex method [5], to improve the efficiency further.

IV. Application to Functional Fitting

In this section, we illustrate the application of the Monte Carlo simulated annealing approach to a functional fitting problem. The specific example we will consider arose in the context of an LCAO (Linear Combination of Atomic Orbitals) electronic bandstructure calculation, but it is typical of a wide variety of fitting problems.

We shall suppose we are given a numerical radial function $V(r)$, which in our case describes the potential energy of electrons a distance r from the nucleus of an atom. In order to make use of analytic identities we would like to express $V(r)$ as a sum of Gaussians:

$$V(r) \approx \sum_{i=1}^{n} c_i e^{-\alpha_i r^2}. \tag{21}$$

We ask for the best least-squares fit by minimizing the error E_0 with respect to $\boldsymbol{\alpha}$ and $\mathbf{c}$, with

$$E_0(\mathbf{c};\boldsymbol{\alpha}) = \sum_j w_j \left[V(r_j) - \sum_{i=1}^{n} c_i e^{-\alpha_i r_j^2} \right]^2 + C_1 \sum_{i=1}^{n} c_i^2. \tag{22}$$

The last term in Eq. (22) is an optional penalty function which can be used to suppress large values of the linear coefficients. Equation (22) can easiy be minimized with respect to the linear coefficients c_i using linear least-squares methods, so that we may think of $E_0 = E_0(\boldsymbol{\alpha})$ with the optimal c_i's always inserted in Eq. (22).

The choice of the optimal set of α_i's is a non-linear global optimization problem, and generally exhibits multiple local minima. We can, of course, just write a subroutine to evaluate $E_0(\boldsymbol{\alpha})$ and implement our Monte Carlo annealing procedure with $E = E_0$ and $\mathbf{x} = \boldsymbol{\alpha}$. To make the procedure somewhat more efficient and flexible, however, we make the following modifications. First, we define the objective function E for the Metropolis algorithm to be

$$E(\boldsymbol{\alpha}) = \ln(E_0) + C_2 \sum_{i=1}^{n} \ln^6(\alpha_i/\alpha_i^{(0)}) + C_3 \sum_{i=2}^{n-1} \left[\frac{\ln(\alpha_i/\alpha_{i-1})}{\ln(\alpha_{i+1}/\alpha_i)} - \frac{\ln(\alpha_{i+1}/\alpha_i)}{\ln(\alpha_i/\alpha_{i-1})} \right]^2. \tag{23}$$

Because E_0 is positive definite, the replacement of E_0 by $\ln(E_0)$ is natural, and it emphasizes the differences between the depths of competing local minima. Two additional optional penalty functions have been included. The first, whose importance is controlled by C_2, can be used to suppress choices of $\boldsymbol{\alpha}$ which deviate greatly from the initial guess. (The form has been chosen so that the penalty function is small for $|\ln \alpha - \ln \alpha^{(0)}| < C_2^{-1/6}$, but rises sharply outside this range; the sharpness is controlled by the exponent, which has been rather arbitrarily set to 6.) The second, controlled by C_3, suppresses choices of $\boldsymbol{\alpha}$ which are poorly even-tempered, i.e., which do not resemble a geometric series. Such penalty functions can play either of two roles: (i) they may reflect real optimization considerations [for example, very large values of the linear coefficients may make the evaluation of $V_{\text{fit}}(r)$ subject to numerical round-off error], or (ii) even if the penalty function is negligible in the vicinity of the minimum, it may inhibit the random walk from searching pointlessly in large volumes of parameter space corresponding to unreasonable solutions. Because the gradients of $E(\boldsymbol{\alpha})$ are not needed, such penalty functions are easily implemented.

Finally, we make the transformation

$$\alpha_i = \sum_{k=1}^{i} e^{x_k} \tag{24}$$

and carry out the random walk in the space of x_k's, minimizing $E(\mathbf{x})$. This orders the decay constants, insures that cases of close approach between an α_i and α_{i+1} are handled in a well-controlled fashion, and reduces the volume of phase space associated with unreasonably large decay constants.

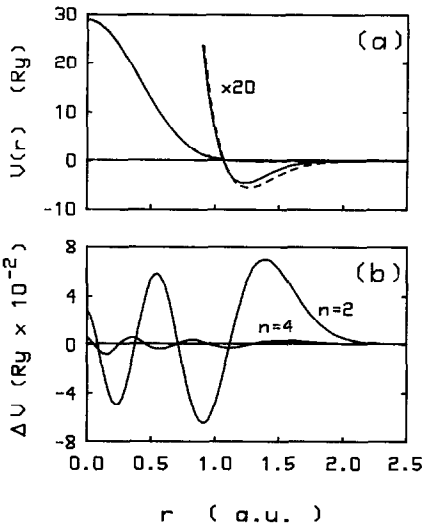

FIG. 3. Fitting of pseudopotential $V(r)$ to a sum of Gaussians. (a) Target function (solid line) and fit (dashed line) for $n = 2$. (b) Difference ΔV between target and fit for $n = 2$ and $n = 4$.

Figure 3 shows the results of applying the Monte Carlo simulated annealing algorithm to such a problem. The numerical function to be fitted is the difference between the potential felt by s and p electrons in carbon, according to a pseudopotential generation scheme of Hamann *et al.* [11]. We have used a uniform radial grid with $\Delta r = 0.1$ a.u. to represent the function values, with uniform weights assigned to each grid point. The fitting parameters were $C_1 = 2 \times 10^{-6}$, $C_2 = 0$, $C_3 = 0.02$, $T_0 = 1$, $\chi_S = 2$, and $\chi_T = 0.8$; these were chosen essentially by trial and error. The fit was carried out with $n = 2$, 4, and 8 decay constants, with the maximum fitting errors being 7×10^{-2} Ry, 9×10^{-3} Ry, 7×10^{-4} Ry, respectively. In order to get a good fit in the chemically important tail region, the $n = 4$ fit was adopted for later use.

Finally, Table III shows that for $n = 4$ the objective function is indeed multiex-

TABLE III

Local Minima for $n = 4$ Fitting Problem, with Number of Arrivals (out of a Sample of 20) Given for Different Values of χ_T

						n_{arriv} for χ_T			
Min	E	α_1	α_2	α_3	α_4	0.2	0.4	0.6	0.8
1	−6.033	8.93	3.76	3.14	2.72	14	13	18	19
2	−4.464	33.45	2.20	1.77	1.68	2	4	1	1
3	−3.937	2.57	1.39	1.07	0.90	4	3	1	0

tremal, with the three deepest local minima identified. Note that $\chi_T > 0.5$ is needed to insure arrival at the global minimum with high probability.

V. Summary and Conclusions

We have shown how the Monte Carlo simulated annealing approach to discrete optimization may be equally well applied to optimization over continuous parameter spaces. The efficiency of this approach is enhanced by employing a self-regulatory mechanism which maintains a random step distribution appropriate to the local topography and current temperature. The method is tested on a set of standard global optimization problems, and its performance is shown to be competitive with the best algorithms currently available. Finally, it is applied to a typical functional fitting problem of the sort which might arise in many branches of physics.

While the method has not been carefully tested for large-dimensional systems ($n \gtrsim 10$), we see no major obstacles to such an application. Physical applications of the Metropolis approach have long been applied to large-n systems, with $n \approx 100$ or more.

In short, our preliminary experience with the Monte Carlo annealing approach to continuous global optimization is encouraging. With or without further development, it may prove to be the method of choice for many classes of problems.

Acknowledgments

Support for this work was provided by NSF Grant DMR7822465 and by a program development fund from the Director of the Lawrence Berkeley Laboratory. We (D.V. and S.G.L.) would also like to acknowledge receipt of Miller Institute and Alfred P. Sloane Foundation fellowships, respectively.

References

1. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science* **220** (1983), 671.
2. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. *J. Chem. Phys.* 21 (1953), 1087.
3. See, for example, G. Dahlquist and A. Bjoerck, "Numerical Methods," Prentice–Hall, Englewood Cliffs, N.J., 1974.
4. L. C. W. Dixon and G. P. Szego, *in* "Towards Global Optimization 2" (L. C. W. Dixon and G. P. Szego, Eds.), pp. 1–18, North-Holland, New York/Amsterdam, 1978.
5. J. A. Nelder and R. Mead, *Comput. J.* 7 (1967), 308.
6. A. A. Torn, *in* "Towards Global Optimization 2" (L.C.W. Dixon and G. P. Szego, Eds.), pp. 49–62, North-Holland, New York/Amsterdam, 1978.
7. J. Gomulka, *ibid.*, pp. 63–70.
8. W. L. Price, *ibid.*, pp. 71–84.
9. J. Mockus, V. Tiesis, and A. Zilinskas, *ibid.*, pp. 117–130.
10. L. de Biase and F. Frontini, *ibid.*, pp. 85–102.
11. D. R. Hamann, M. Schluter, and C. Chiang, *Phys. Rev. Lett.* **43** (1979), 1494.