Weighted-ensemble simulated annealing: Faster optimization on hierarchical energy surfaces

Gary A. Huber* and J. Andrew McCammon
Department of Chemistry and Biochemistry, Department of Pharmacology, University of California, San Diego, La Jolla, California 92093-0365
(Received 31 October 1996)

A method, weighted-ensemble annealing, is proposed for finding the global minima of complicated functions, such as those found in biological problems. This method performs simulated annealing using multiple system copies; it automatically adjusts the distribution of copies and the allocation of computer resources as the cooling proceeds. This readjustment procedure is designed to take advantage of the hierarchical structure of the energy landscape of biomolecules and other systems. This method is applied to a fractal-like function with energy barriers of many sizes and a large entropy barrier. It is shown that using an optimal number of system copies results in a success rate for finding the global minimum, which is an order of magnitude higher than the success rate from traditional single-copy annealing, using the same total number of function evaluations. [S1063-651X(97)06804-9]

PACS number(s): 02.70.Lq, 02.60.Pn, 36.20.Hb, 87.15.By

I. INTRODUCTION

One of the most important general computational challenges is to find the global minimum of a complicated multidimensional function. Many problems in biochemistry are equivalent to finding a global potential energy minimum, such as protein folding and ligand binding. Such problems also arise in materials science and other areas. This is a difficult problem, because the potential energy surface contains many local minima separated by high barriers, which often defeat minimization algorithms. In describing the following algorithms, it is helpful to think of the domain of the function variables as a multidimensional *configuration space*, and each set of variable values as a *system copy*. The process of changing the variable values, or moving a system copy through configuration space, is called a *move*.

One approach to the global minimum problem is *simulated annealing* (SA) [1], in which a temperature parameter is defined, set to a high value, and gradually lowered as a single system copy is subjected to moves. The move set is defined so that large energy barriers can be surmounted at high temperatures, while at low temperatures, the system settles in one minimum. The more slowly the temperature decrease, or *cooling*, proceeds, the more likely is the chance of finding the global minimum. SA has been successfully used in problems of combinatorial optimization, such as the ''traveling salesman'' problem and the optimal layout of electronic chips [2].

Unfortunately, SA is not very effective for models of protein folding [3,4]. The reason is that the system often settles into a local minimum that is not the global minimum, and it is then unable to escape. Thus, extremely long simulations with extremely slow cooling would be necessary to find the global potential energy minimum with a high probability. Straub [5] presents a simple argument showing that the required time will scale exponentially with the difference between these two quantities: the largest barrier height between the global minimum and another minimum, and the smallest energy difference between the global minimum and another minimum. This difference is large in proteins.

Fortunately, the potential energy landscape of proteins has another feature that opens the possibility to other minimization algorithms. There is evidence, both theoretical [6] and experimental [7], that the energy landscape has a *hierarchical* form. Such functions have a component consisting of large energy variations and large length scales, along with a hierarchy of other components with smaller length scales and correspondingly smaller energy variations. In addition, the global minimum appears to sit inside a nested set of basins. Each basin contains the global minimum of an energy function that would result from subtracting all components with a length scale smaller than a given value (in other words, smoothing the function). So, the most promising approach is to locate the outermost basin on the coarsest scale, and successively locate the inner basins surrounding the global minimum. Several schemes have been used that smooth the energy function [8,9], as well as those which ''blur'' the system copy [10,11]. The effectiveness of these methods for finding the global minimum in peptides [12–14] and simple models of proteins demonstrates the effectiveness of this strategy.

In this paper, we describe a scheme, weighted-ensemble annealing (WEA), that uses multiple copies of the system, which are moved independently of each other. As the temperature T is lowered, some of the copies become trapped in higher-energy minima; these are deleted, and the copies that appear to proceed towards promising regions of configuration space are duplicated, focusing the simulation on the current basin surrounding the global minimum. The total number of copies remains constant. Other SA schemes with multiple copies have been used on combinatorial optimization problems; the main goal of these schemes has been the adaptation of simulated annealing to parallel computation [15–17], or the gathering of auxiliary information about the energy surface itself [18].

*Author to whom correspondence should be addressed.

II. METHOD

In the weight-ensemble annealing method, each copy of the system is given a statistical weight. This is similar to the treatment of multiple copies in the weighted-ensemble Brownian dynamics of Huber and Kim [19]. As the cooling progresses, the weights are adjusted and used as a criterion for deleting and duplicating copies.

At the beginning of the annealing, the N system copies are scattered randomly and uniformly throughout configuration space, and each copy is given a weight proportional to its Boltzmann factor $\exp(-V/kT)$. The total weight of the ensemble is then normalized to N. This procedure provides an approximation to a Boltzmann distribution by the ensemble of copies.

The annealing process comprises the following steps. First, the copies are moved, with moves being accepted or rejected as above. Next, the temperature is lowered a small amount, depending on the annealing schedule. Finally, the weights of each copy are adjusted according to the following formula:

$$p_i \leftarrow p_i \exp\left[V_i \left(\frac{1}{T_n} - \frac{1}{T_{n+1}} \right) \right], \tag{1}$$

where p_i is the weight of copy i, its energy is V_i, the temperature at the current step n is T_n, and T_{n+1} is the new temperature. The weights of the ensemble are then normalized to N. The important effect is that the weights of the low-energy copies increase, while the copies in higher-energy regions ''fade away.'' In the actual implementation, the logarithms of the weights, rather than the weights themselves, are stored and manipulated in order to avoid rounding errors.

Equation (1) is used for the following reason. Let us divide up the configuration space into separate *states*. For discrete models, where the configuration variables take discrete values, this is trivial. For continuous models, the configuration space can be subdivided into hypercubes that are small enough that the energy is almost constant within. Suppose that N is very large so that each state is occupied by a very large number of copies. Suppose also that the copies are distributed according to a Boltzmann distribution at T_n. Then, the total copy weight in state i is proportional to $\mathcal{V}_i \exp(-V_i/T_n)$, where $\mathcal{V}_i$ is the volume of the state. If the temperature is dropped to T_{n+1}, then the weights in state i are multiplied by the factor in Eq. (1). The total weight in i is now $\mathcal{V}_i \exp(-V_i/T_{n+1})$, which satisfies the Boltzmann distribution at T_{n+1}. Heuristically, having the distribution be as close as possible to a Boltzmann distribution at the current temperature is desirable for an annealing algorithm.

As the annealing proceeds and the weights are adjusted, eventually some of the copies become trapped in high-energy regions and their weights become very small. It makes little sense to use computer resources in moving them, so a method is needed for removing them, without disrupting the Boltzmann distribution. As the same time, the weights of other copies become larger, and it makes sense to split those copies, in order to keep the number N constant and to focus the search for the minimum. At regular intervals, the weights are sorted. If the largest weight is greater than two times the smallest weight, the copy with the largest weight is duplicated, and the copies with the two smallest weights are combined. The weights are sorted again, and the procedure is repeated until the largest weight is less than twice the smallest. This is similar in spirit to genetic algorithms, where the ''unfit'' copies are eliminated and the ''fit'' copies replicate [20–22].

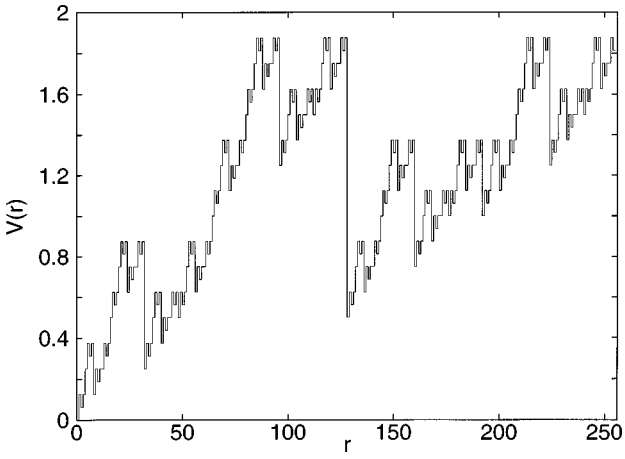

FIG. 1. Fractal-like function.

When a copy is split, a duplicate is created with the same position in configuration space, but the parent and duplicate now each have half of the original parent's weight. When two copies i and j are combined, one of the copies is deleted, and its weight is added to the weight of the other copy. To choose which copy to delete, a random number between 0 and 1 is generated. If it is less than $p_i/(p_i+p_j)$, then copy j is deleted; otherwise, copy i is deleted. The surviving copy is left with weight p_i+p_j. As shown in Appendix A of Huber and Kim [19], this procedure introduces no bias and preserves the Boltzmann distribution.

III. FRACTAL FUNCTION

In order to test the WEA algorithm, we construct a multidimensional function that has high energy and entropy barriers, yet has a hierarchical structure that one might find in a fractal or the energy landscape of a protein. There are four variables x_1 through x_4, and the potential energy takes the form $V(r)$, where $r=\sqrt{x_1^2+x_2^2+x_3^2+x_4^2}$ is the Euclidean distance from the origin in the four-dimensional configuration space. The function $V(r)$ is shown in Fig. 1. It is constructed by starting with a simple motif and by adding smaller versions of itself to the original. This is repeated to create four self-similar levels. The variables are constrained to lie between -256 and 256; the energy is assumed infinite for $r>256$. The energy barriers are evident from the picture. The entropy barrier arises from the fact that the total volume of the accessible configuration space, a four-sphere with a radius of 256, is $0.5\pi^2\times 256^4 = 2.12\times 10^{10}$, while the volume of the space representing the global minimum, a four-sphere with a radius of 1, is only $0.5\pi^2 = 4.93$. In the following series of annealing runs, the moves consisted of displacements drawn from a spherically symmetric Gaussian distribution with a variance of 16.0, and were accepted or

TABLE I. Frequency of success in finding the global minimum during 20 runs for each number of copies and total number of moves.

	No. of copies													
	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
No. of moves														
1 048 576	20	20	20	20	20	20	18	16	13	8	7	5	4	1
524 288	20	20	20	19	20	19	15	16	14	9	5	3	2	1
262 144	19	19	18	17	19	15	15	11	2	8	8	0	1	0
131 072	12	13	15	16	15	10	13	6	7	4	0	2	0	1
65 536	2	8	8	9	6	4	8	5	4	0	2	1	1	0
32 768	0	0	1	2	4	8	7	2	0	2	2	0	0	0
16 384	0	1	0	2	2	2	2	1	1	1	0	0	0	0
8 192	0	1	0	0	0	0	1	0	0	1	0	0	0	0
4 096	0	0	0	0	0	0	1	0	1	0	0	0	0	0
2 048	0	0	0	0	0	0	0	0	0	0	0	0	0	0

rejected according to the Metropolis Monte Carlo algorithm [23].

For this system, a series of annealing runs were performed, varying both the number of copies and the total number of moves for each run. The starting temperature is $2/k$ and the final temperature is $0.005/k$, where k is Boltzmann's constant. At $kT=0.005$, a system copy drawn from a thermal equilibrium distribution has a chance greater than 90% of being at the global minimum $V=0$. This presents a severe challenge for an annealing algorithm. The system must be cooled to a low temperature, but the low temperature makes it easy for the system to be trapped in other minima. An exponential cooling schedule was used, using the following updating formula for T:

$$T_{n+1}=\alpha T_n, \tag{2}$$

where the constant α is chosen so that the final T is reached in the prescribed number of moves.

A total number of 2800 runs was performed. The number of copies was varied from 1 to 8192, increasing by a factor of 2 each time, and the total number of moves was varied from 2048 to 1 048 576, increasing by a factor of 2 each time. The total number of moves takes into account the number of copies, so a run with two copies is cooled twice as quickly as a run with one copy, for the same number of moves. For each pair of copy numbers and move numbers, 20 runs were performed; a run was declared successful if at least one copy reached the global minimum at least once during the annealing. For the cases where the number of copies exceeded the total number of moves, all runs were declared unsuccessful. We use the total number of moves as a rough measure of computer time required; for more complex systems, most of the computer time will be consumed in computing the moves.

IV. RESULTS

The success rates from the fractal function runs are summarized in Table I. Note that each horizontal row of numbers required the same total number of moves for each entry. It is apparent that an order-of-magnitude improvement in the success rate for finding the global minimum is possible with a good choice of the number of copies. It appears that the optimal choice for the number of copies depends on how many moves one is will to take; the higher the number of moves, the more copies should be used. This, perhaps is due to a trade-off between local searching, which is emphasized by fewer copies, and global searching, which is emphasized by more copies. Of course, the optimal number for a given model cannot yet be known ahead of time, but future research might point the way towards some guidelines.

The success of the WEA method on this easily evaluated yet challenging function indicates some promise for biological problems. Other models currently being studied are the lattice model of protein folding previously studied by Socci and Onuchic [24], and the tetra-alanine peptide in a vacuum by means of molecular dynamics [25].

This method also has promise for parallel computing, where each processor could handle a system copy. Communication between processors would be limited to comparing statistical weights (frequent communication, but very little data) and copy deletions and duplications (much data, but infrequent communication). Thus, the WEA method might run very efficiently on a parallel computer, because the communication bottlenecks might be small [26].

Other advances in global optimization can be used under the framework of WEA. There has been much work in the area of more efficient move sets that seek to avoid repeatedly searching the same region of configuration space, usually by surmounting or circumventing energy barriers [27–29] or by ''remembering'' past conformations [30,31]. Any advances made in this area are directly complementary to the WEA method, because any move set can be used in WEA. Also, some of the methods mentioned above that blur the potential energy surface or the system copy itself could be used to facilitate configuration space exploration.

ACKNOWLEDGMENTS

This work has been supported in part by NIH, NSF, and the MetaCenter Program of the NSF Supercomputer Centers. G.A.H. would like to thank NIGMS for financial support.

[1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).
[2] I. H. Osman and G. Laporte, Ann. Oper. Res. **63**, 513 (1996).
[3] R. C. Brower, G. Vasmatzis, M. Silverman, and C. Delisi, Biopolymers **33**, 329 (1993).
[4] A. Nayeem, J. Vila, and H. A. Scheraga, J. Comput. Chem. **12**, 594 (1991).
[5] J. E. Straub, in *New Developments in Theoretical Studies of Proteins*, edited by R. Elber (World Scientific, Singapore, 1996).
[6] P. Leopold, M. Montal, and J. N. Onuchic, Proc. Natl. Acad. Sci. U.S.A. **89**, 8721 (1992).
[7] H. Frauenfelder, S. Sligar, and P. G. Wolynes, Science **254**, 1598 (1991).
[8] L. Piela, J. Kostrowicki, and H. A. Scheraga, J. Phys. Chem. **93**, 3339 (1989).
[9] D. Shalloway, Global Optimiz. **2**, 281 (1992).
[10] J. Ma, D. Hsu, and J. E. Straub, J. Chem. Phys. **99**, 4024 (1993).
[11] G. Verkhivker, R. Elber, and W. Nowak, J. Chem. Phys. **97**, 7838 (1992).
[12] A. Roitberg and R. Elber, J. Chem. Phys. **95**, 9277 (1991).
[13] P. Amara and J. E. Straub, J. Phys. Chem. **99**, 14 840 (1995).
[14] J. Kostrowicki and H. A. Scheraga, J. Phys. Chem. **96**, 7442 (1992).
[15] G. Ruppeiner, J. M. Pedersen, and P. Salamon, J. Phys. **1**, 455 (1991).
[16] R. Azencott, *Simulated Annealing: Parallelization Techniques* (Wiley, Chichester, 1992).
[17] P. Rujan, Z. Phys. B **73**, 391 (1988).
[18] R. Frost, N. Priebe, and P. Salamon, in *An Ensemble-Based Simulated Annealing Library for Multiprocessors*, in ISUG-93, St. Louis, MO, 1993, edited by N. Branan (Intel Supercomputer Systems, Portland, OR, 1993).
[19] G. Huber and S. Kim, Biophys. J. **70**, 97 (1996).
[20] R. S. Judson, E. P. Jaeger, A. M. Treasurywala, and M. L. Peterson, J. Comput. Chem. **14**, 1407 (1993).
[21] R. Unger and J. Moult, J. Mol. Biol. **231**, 75 (1996).
[22] P. Tuffery, C. Etchebest, S. Hazout, and R. Lavery, J. Comput. Chem. **14**, 790 (1993).
[23] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[24] N. D. Socci and J. N. Onuchic, J. Chem. Phys. **101**, 1519 (1994).
[25] I. Andricioaei and J. E. Straub, Phys. Rev. E **53**, R3055 (1995).
[26] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms* (Benjamin-Cummings, Redwood City, CA, 1994).
[27] D. Byrne, J. Li, E. Platt, B. Robson, and P. Weiner, J. Comput.-Aided Mol. Des. **8**, 67 (1994).
[28] R. C. van Schaik, W. F. van Gunsteren, and H. J. C. Berendsen, J. Comput.-Aided Mol. Des. **6**, 97 (1992).
[29] A. Elofsson, S. M. L. Grand, and D. Eisenberg, Proteins **23**, 73 (1995).
[30] T. Huber, A. E. Torda, and W. F. van Gunsteren, J. Comput.-Aided Mol. Des. **8**, 695 (1994).
[31] B. Mehlig, D. W. Heermann, and B. M. Forrest, Phys. Rev. B **45**, 679 (1991).