Cool Walking: A New Markov Chain Monte Carlo Sampling Method

SCOTT BROWN, TERESA HEAD-GORDON
Department of Bioengineering, University of California, Berkeley and Physical Biosciences Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720

Received 5 March 2002; Accepted 12 August 2002

Abstract: Effective relaxation processes for difficult systems like proteins or spin glasses require special simulation techniques that permit barrier crossing to ensure ergodic sampling. Numerous adaptations of the venerable Metropolis Monte Carlo (MMC) algorithm have been proposed to improve its sampling efficiency, including various hybrid Monte Carlo (HMC) schemes, and methods designed specifically for overcoming quasi-ergodicity problems such as Jump Walking (J-Walking), Smart Walking (S-Walking), Smart Darting, and Parallel Tempering. We present an alternative to these approaches that we call Cool Walking, or C-Walking. In C-Walking two Markov chains are propagated in tandem, one at a high (ergodic) temperature and the other at a low temperature. Nonlocal trial moves for the low temperature walker are generated by first sampling from the high-temperature distribution, then performing a statistical quenching process on the sampled configuration to generate a C-Walking jump move. C-Walking needs only one high-temperature walker, satisfies detailed balance, and offers the important practical advantage that the high and low-temperature walkers can be run in tandem with minimal degradation of sampling due to the presence of correlations. To make the C-Walking approach more suitable to real problems we decrease the required number of cooling steps by attempting to jump at intermediate temperatures during cooling. We further reduce the number of cooling steps by utilizing "windows" of states when jumping, which improves acceptance ratios and lowers the average number of cooling steps. We present C-Walking results with comparisons to J-Walking, S-Walking, Smart Darting, and Parallel Tempering on a one-dimensional rugged potential energy surface in which the exact normalized probability distribution is known. C-Walking shows superior sampling as judged by two ergodic measures.

 J Comput Chem 24: 68–76, 2003

Key words: Metropolis Monte Carlo; ergodicity; detailed balance; simulation; sampling

Introduction

In standard Metropolis Monte Carlo (MMC) simulations the exploration of configuration space occurs by proposing random single-particle trial moves that advance the current configuration to nearby (local) points in configuration space.[1] Each new candidate configuration is accepted with probability $\min[1, \exp(-\beta\Delta V)]$; where $\beta = 1/k_B T$ and ΔV is the difference in potential energy between the current and candidate configurations. Using this acceptance criterion, a long sequence of local updates will produce a set of configurations that are weighted according to the Boltzmann distribution. For this method to be efficient the system of interest must be able to surmount potential energy barriers separating the minima on the energy surface, thereby sampling configurations that span a representative portion of thermally accessible states. In situations where barrier heights can be large compared to thermal energy, as in difficult systems like proteins or spin glasses, standard MMC sampling is inadequate and may result in time scales for convergence of statistical averages that can exceed computational limitations. This inability of standard MMC to traverse regions of low probability in configuration space leads to so-called "quasi-ergodicity".

To overcome the problem of quasi-ergodicity, it is necessary to seek alternatives in generating trial moves. For a trial move to realize a non-negligible possibility of crossing energy barriers in the system, the candidate configurations must necessarily be sampled from a larger surrounding region of configuration space, and therefore trial moves must involve nonlocal updates such as the collective displacement of all the atoms in the system. One way to implement N-particle updates is to perform independent local displacements for each individual atom; however, for a condensed

Correspondence to: T. Head-Gordon

Contract/grant sponsor: UC Berkeley

Contract/grant sponsor: LBNL DOE/LDRD

phase system this procedure produces energy changes that are too large, and it has been shown to result in prohibitively low acceptance rates.[2]

Many approaches seeking to enhance the rate of exploration of configuration space have been proposed; of particular relevance to this work are the methods of force-bias Monte Carlo,[3] smart Monte Carlo,[4] and extensions of these methods such as Hybrid Monte Carlo (HMC).[5] HMC generates *N*-particle moves by using constant energy Molecular Dynamics (MD). Starting from an initial configuration, the momenta are sampled from a Maxwell distribution and the configuration is advanced forward in time for some number of MD steps. The resulting candidate configuration is then accepted or rejected based on a Metropolis criterion for the Hamiltonian of the system. In this way a small number of MD steps are able to produce a trial move that involves collective displacement of all the atoms in the system, while avoiding low acceptance rates. HMC offers a substantial improvement over standard MMC sampling; however, it has one drawback stemming from the fact that resampling the momenta at the start of each HMC step is akin to a diffusive exploration of phase space, the dynamics of which can lead to a decrease in the rate of barrier crossing.[6]

One strategy for dealing with this is to combine HMC with the method of Jump Walking (J-Walking).[7] With J-Walking the sequence of HMC steps is periodically interrupted by attempts to jump to configurations sampled from a higher temperature distribution. This is implemented by running two separate "walkers" at two different temperatures. One walker is run at the temperature of interest, while the other walker is run at a sufficiently high temperature such that barrier crossing is not problematic. The low-temperature walker is periodically updated with a configuration obtained from the high-temperature distribution that occupies a much larger region of configuration space. This enhances the rate of exploration of configuration space by facilitating moves between minima on the surface, which circumvents the problem of quasi-ergodicity found at low temperatures. A potential problem with J-Walking is that it does not strictly satisfy detailed balance, as the jump transition probabilities do not (in general) generate a true Markov process.[8,11,17] This is not to say that results from J-Walking simulations will necessarily be poor, particularly for large configuration sets at the high temperature.

An inherent difficulty with the two-stage J-Walking construction is that the likelihood of performing successful jumps is very low due to the small overlap between the two distributions at the high and low temperatures. To overcome this difficulty one can implement multiple-stage J-Walking[9] or Parallel Tempering,[10,11] both of which use a number of walkers at temperatures intermediate between the high and low-temperature walkers. The intermediate walkers are more closely spaced with regards to temperature and so presumably possess a greater degree of overlap between neighboring distributions, which is conducive to higher jump-acceptance probabilities. This approach does result in substantially higher jump acceptance. However, it may be too demanding in terms of computational cost for any system with a physically realistic level of complexity, although the method is well suited and manageable on a parallel computing platform.

In Parallel Tempering jump moves involve the exchange of configurations between neighboring temperatures. This swapping of configurations couples the Markov chains at different temperatures, and it can be shown to rigorously produce a true Markov chain process. Parallel Tempering is more desirable than multiple-stage J-Walking, as its results can be more straightforwardly analyzed since it can be rigorously shown to give the correct limiting distribution.

Finally, we should mention here the sampling scheme known as Simulated Tempering,[12,13] from which the method of Parallel Tempering originates. Simulated Tempering is similar to Parallel Tempering in that a parameter space, that is, the temperature, is explored in addition to configuration space. The difference with Simulated Tempering is that the acceptance criterion for moves between temperatures requires knowledge of normalizing factors. These factors may require some initial trial and error estimation during preliminary runs in order to achieve optimal swap acceptance between neighboring temperatures.

A different approach that attempts to circumvent the necessity of using multiple walkers is Smart Walking (S-Walking).[14] In S-Walking a trial configuration is sampled from the high-temperature distribution, and then steepest-decent is quenched before evaluating any jump-acceptance criterion. The quenching process lowers the potential energy of the trial configuration, bringing its energy close to that of a local minimum. Consequently, the energy of the quenched structure is more likely to be consistent with typical energies of the structures present in the low temperature distribution. The method is successful in that it does improve the rate of sampling of configuration space. Additionally, S-Walking significantly reduces the computational cost compared to that of multiple-walker simulations; however, it does not formally satisfy detailed balance, and therefore gives no guarantee that a Boltzmann distribution will be produced at the low temperature. To obtain quantitative results from the S-Walking methodology it is necessary to restrict the frequency of jump attempts in order to minimize the error introduced by sampling from a non-Boltzmann distribution. The reduction of jump frequency has the added effect of slowing the convergence because this directly influences how fast the walker is able to hop between basins.

S-Walking can be corrected to satisfy detailed balance using a strategy called Smart Darting.[15] In Smart Darting one catalogues the positions of all minima of interest, and uses the positions to create a set of "darts." Each dart is a displacement vector between minima, and thus for N minima one has $N(N-1)$ darts. Jump moves between minima are only attempted when the current low-temperature configuration lies within a small region surrounding one of the (cataloged) minima. Smart Darting corrects the acceptance probability for S-Walking type moves such that microscopic reversibility is restored, although for highly complicated surfaces the management of a comprehensive catalogue of a very large number of minima may become prohibitive.

In this article we present a new approach to the sampling of rough energy surfaces that we call Cool Walking (C-Walking), which pools the best features among the methods of J-Walking, S-Walking, and Parallel Tempering. In C-Walking, simulations are performed with two walkers run in tandem. One walker is run at a high temperature and the other walker at a low temperature. The high-temperature walker is used to generate nonlocal candidate moves for the low-temperature walker. These nonlocal moves are constructed by first sampling a configuration from the high-temperature walker and then performing a statistical quenching pro-

cess on the sampled configuration. This is similar in spirit to the quenching step in S-Walking, yet it satisfies a detailed balance equation. As is done in Parallel Tempering, we attempt configuration swapping moves between states at the cooled and low temperatures. To improve the efficiency of this process we attempt to perform these swaps during the cooling process. Further improvement on efficiency can be realized by performing these jumps between "windows" of states.

C-Walking has greater jump-acceptance probabilities, rigorously satisfies detailed balance, and, in terms of computational efficiency, is at least several times more efficient in sampling configuration space than J-Walking, S-Walking, Smart Darting, or Parallel Tempering for the simple but fully characterized model we investigate.

In the Methods section we present the details of the types of jump moves used in J-Walking, S-Walking, Smart Darting, Parallel Tempering, and C-Walking. Then in the Results section we present a comparison of the five methods with data obtained from simulations on a one-dimensional (1-D) potential energy surface. As we can use exact ergodic measures for this surface, the simulations provide a clear measure of the relative performance of each method. In the final section we conclude with a summary of these results and a discussion of future directions.

Methods

We wish to compare the rates at which J-Walking, S-Walking, Smart Darting, Parallel Tempering, and C-Walking sample configuration space. As was done previously,[14] we perform our comparison between the methods by sampling from a 1-D potential. Although more complex systems provide better tests for a method's ability to sample ergodically, they also pose a greater difficulty in analyzing the sampling efficiency. On a 1-D rough potential energy surface we can define exact ergodic measures, while on a high-dimensional rough energy surface we can only postulate measures of ergodicity. Thus it makes sense to perform initial trials on a simple, well-characterized system.

The potential energy function used for the 1-D system is

$$V(x) = \sum_{n=1}^{20} C_n \sin\left(\frac{2n\pi x}{L}\right) \tag{11}$$

where the coefficients $\{C_n\}$ are chosen on the interval $[-1, 1]$, and the length of the simulation box in reduced units is $L = 10$ (note that the units for energy are arbitrary). The coefficients used here are given in Table I. A plot of the potential energy function is shown in Figure 1, along with the exact probability distribution functions at the reduced temperatures of the high- and low-temperature walkers, $T^* = 3.0$ and 0.1, respectively. The exact distribution functions are calculated explicitly from the expression $\rho_{\text{exact}}(x) = \exp(-\beta V(x))/Z$; where Z is the configuration integral. We use 1000 data points for collecting $\rho(x)$ and for comparing to $\rho_{\text{exact}}(x)$.

We note that in both the original J-Walking[7] and S-Walking[14] articles the 1-D surfaces used were rather smooth, having only a single dominant minimum. The surface used here has four minima with significant population at the low temperature. While the high temperature sees the four lowest minima roughly equally populated, at the low temperature we find that the relative weighting becomes distinctly unequal. This provides a sensitive test as to the quality of sampling with each method. On several occasions we were able to identify subtle flaws in our algorithm based solely on its performance on this surface. The ability to reproduce the relative weighting for the populated minima at $T^* = 0.1$ is fairly nontrivial for this surface.

Table 1. Coefficients for the One-Dimensional Potential Energy Function.

C_1	−0.466516	C_{11}	0.891462
C_2	−0.834376	C_{12}	−0.665239
C_3	−0.714529	C_{13}	0.810546
C_4	−0.0245586	C_{14}	0.198216
C_5	0.238837	C_{15}	−0.816637
C_6	0.0143649	C_{16}	−0.195351
C_7	0.271003	C_{17}	−0.573181
C_8	−0.374538	C_{18}	0.251745
C_9	0.873564	C_{19}	0.647615
C_{10}	−0.370258	C_{20}	0.201654

Markov Chain Monte Carlo (MCMC) refers to a class of methods for generating a set of configurations with weighting proportional to a chosen probability function. To guarantee that the desired limiting distribution is approached, it is sufficient to enforce the condition of detailed balance:

$$P(x)T(x \rightarrow y) = P(y)T(y \rightarrow x) \tag{1}$$

where $T(x \rightarrow y)$ is the transition probability of reaching state y from state x, and $P(x)$ is the probability of realizing state x. If a new state y is generated from a proposal distribution function $T(\cdot)$, then the acceptance criteria for the new state is given by

$$\text{acc}(x \rightarrow y) = \min\left[1, \frac{P(y)T(y \rightarrow x)}{P(x)T(x \rightarrow y)}\right] \tag{2}$$

The specific form of the jump transition probability $T_J(\cdot)$ is what serves to differentiate the sampling techniques in J-Walking, S-Walking, Smart Darting, Parallel Tempering, and C-Walking. In our implementation of these methods we use HMC moves to regularly update configurations, but occasionally the HMC sequence is interrupted by jump attempts that are drawn from different distributions depending on the particular method. The rate at which jump attempts occur is controlled by a jump probability parameter, P_J. P_J is set equal to 3% for our comparison between the methods. For Parallel Tempering we set $P_J = 3\%$ for each walker, independent of the total number of walkers. The computational cost of J-Walking and Parallel Tempering is roughly constant with P_J, but S-Walking, Smart Darting, and C-Walking have costs that depend on P_J when the two walkers are run in tandem. Thus for the higher jump rates, the methods of S-Walking,

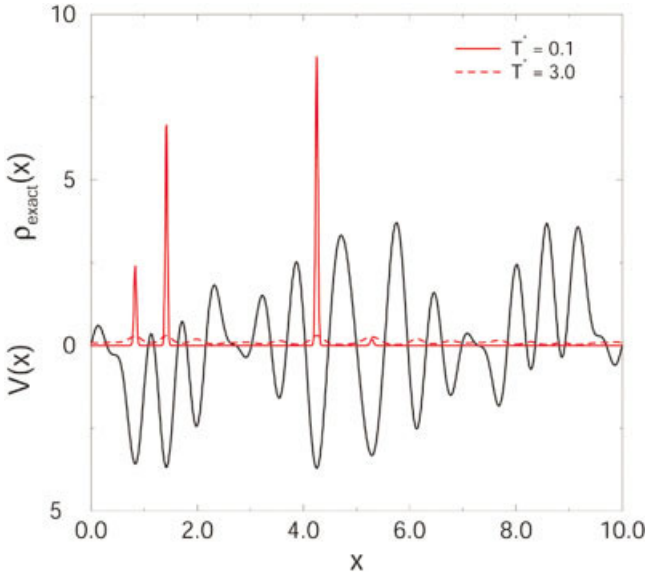

Figure 1. One-dimensional potential energy $V(x)$ versus position x for box length $L = 10$. Also shown in the plot is the ideal distribution function $\rho_{\text{exact}}(x)$ for temperatures of $T^* = 0.1$ and 3.0. The units for the potential energy are arbitrary. At $T^* = 0.1$ the distribution is sharply peaked about the four lowest minima, whereas at $T^* = 3.0$ the peak heights for these minima are roughly equal.

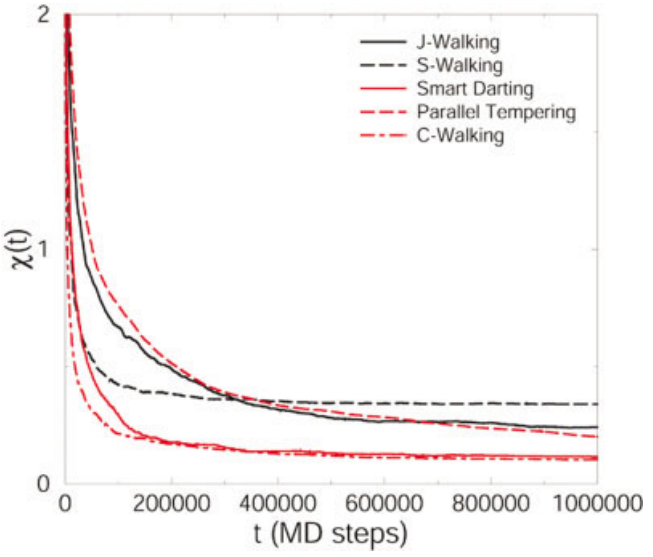

Figure 3. Ergodicity measure $\chi(t)$ versus t (in units of the number of MD steps performed by the low-temperature walker) for J-Walking, S-Walking, Smart Darting, Parallel Tempering, and C-Walking. All simulations were performed with $P_J = 3\%$. The Parallel Tempering results shown are for one intermediate walker.

Smart Darting, and C-Walking have an increased computational cost.

For the J-Walking, S-Walking, Smart Darting, and C-Walking methods two walkers are required, one at a high temperature and another at a lower temperature (the temperature of interest). The purpose of the high-temperature walker is to produce configurations for use in jump moves that provide updates to configurations at the low temperature. Parallel Tempering is distinguished by

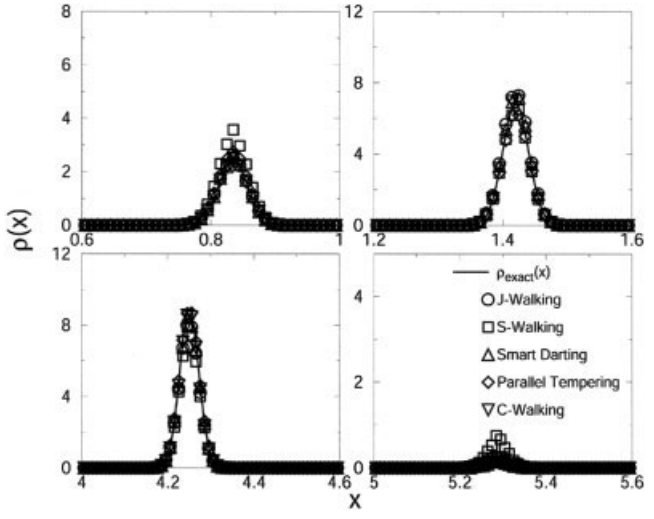

Figure 2. $\rho_{\text{exact}}(x)$ and $\rho(x)$ for $T^* = 0.1$ versus position x for J-Walking, S-Walking, Smart Darting, Parallel Tempering, and C-Walking. All simulations were performed with jump probability $P_J = 3\%$. Only the peaks for the four populated minima at $T^* = 0.1$ are shown. The Parallel Tempering results shown are for one intermediate walker.

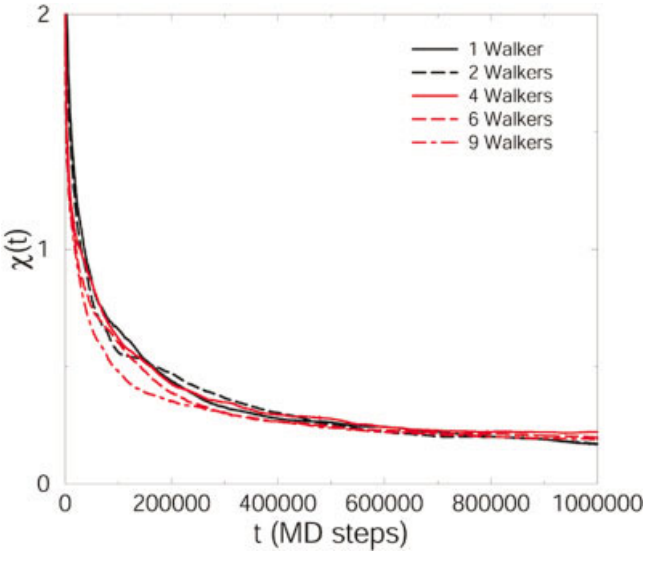

Figure 4. Ergodicity measure $\chi(t)$ versus t (in units of the number of MD steps performed by the low-temperature walker) for Parallel Tempering showing the effects of an increasing number of walkers at intermediate temperatures. All runs have $P_J = 3\%$ between pairs of walkers.

having additional walkers at temperatures intermediate between the target (low) and ergodic (high) temperatures.

The most convenient way to implement the methods involving two walkers at different temperatures is to simultaneously perform all simulations, updating the high- and low-temperature configurations independently and concurrently for each HMC cycle. A potential drawback to running the walkers in tandem is the presence of correlations between successive configurations in the high-temperature walker, which can introduce systematic error into calculated averages at the low temperature. The typical solution (on a single processor platform) is to run the high-temperature walker beforehand and write out configurations to an external file. These configurations can then be randomly chosen from the file during a run of the low-temperature walker. The trade off with this is that storage requirements for the high-temperature configurations will become exceedingly large, cumbersome, and impractical for any complex system with a large number of degrees of freedom. For Smart Darting it is a requirement to run the high-temperature walker beforehand, as only then will one be able to create the required set of darts between the quenched structures at the cataloged minima.

In all the simulations reported here, we run the walkers in tandem, even in the case of Smart Darting. The main reason for doing this is because for any real system of interest, running walkers in tandem is the only practical solution. Furthermore, the organization of the C-Walking scheme lends itself to naturally breaking up correlations in the high-temperature walker, although like the other methods we can write out high-temperature configurations and sample them randomly for C-walking. Thus, propagating the walkers in tandem for all methods allows us to better compare results, as well as allowing us to emphasize an important advantage of C-Walking. To deal with correlations in the J-Walking, S-Walking, and Smart Darting simulations we perform an extra 100 HMC steps (500 MD steps) on the high-temperature walker for each HMC cycle of the low-temperature walker. As jumping is attempted infrequently, we have confirmed that this is equivalent to writing out configurations and sampling them randomly to ensure that configurations between jumps are sufficiently uncorrelated. In Parallel Tempering we do not perform any additional HMC moves.

J-Walking, S-Walking, Smart Darting, and Parallel Tempering

In J-Walking the jump moves occur by sampling a high-temperature configuration and attempting to use it as the next update in the low-temperature Markov chain. Therefore the trial sampling distribution T_J is the probability of finding any particular configuration in the high-temperature walker, namely $T_J(x \rightarrow y) = \exp[-\beta V(y)]/Z_J$; where $V(y)$ is the potential energy of the high-temperature state y and Z_J is the configuration integral at the high temperature. Jump moves are accepted with probability

$$\mathrm{acc}(x \rightarrow y) = \min[1, \exp(-(\beta - \beta_J)[V(y) - V(x)])] \quad (3)$$

As stated previously, this process does not *rigorously* satisfy detailed balance. For finite Markov chains, not every state sampled from the high-temperature walker is guaranteed to be present in the low-temperature Markov chain, and vice versa. For example, if we sample a low-temperature state x (*not* present in the Markov chain at the high temperature) and then sample y from the high-temperature distribution, the probability of choosing x and jumping to y, $T(x \rightarrow y)$, is finite, but not equal to the probability of choosing y and jumping to x, $T(y \rightarrow x)$, which is identical to zero because x is not a member of the high-temperature set. Rigorously speaking, only for infinite Markov chains will detailed balance be satisfied for all x and y in the J-Walking scheme.[8,11,17]

In the case of S-Walking the jumps occur by taking the selected configuration from the high- temperature walker, and subsequently performing an approximate steepest-decent quench to a local minimum. The quenching process relaxes the structures, bringing their energies to values more likely to be representative of the low temperature distribution. The quenched configuration is used as the next trial move at the low temperature. Moves are accepted with probability

$$\mathrm{acc}(x \rightarrow y) = \min[1, \exp(-\beta[V(y) - V(x)])] \quad (4)$$

S-Walking increases the probability for successful jumps with minimal increase in computational cost (about 50% more computation is required relative to two-stage J-Walking[14]); however, as previously stated, S-Walking only approximately satisfies detailed balance. To reduce the error from this, one must select a sufficiently low jump rate such that there is ample time for the low-temperature walker to relax within its local basin prior to attempting subsequent jumps. The time required for this is in the order of the time scale for the decay of energy correlations within the system. In low-temperature systems with rough energy surfaces, these correlations may persist for very long times. To deal with this in systems with slowly decaying energy correlations, it may be necessary to implement extremely low jump rates. It has been shown that the approximate nature of the S-Walking algorithm can lead to substantially skewed results under certain conditions.[15]

Smart Darting modifies the S-Walking procedure by regulating the jumps to trial (quenched) configurations. Before a jump move is allowed additional checks are performed. The current low-temperature configuration is checked to see whether it is near one of a previously cataloged set of potential energy minima. Because the walkers are run in tandem, the next check we perform is to see if the quenched high-temperature configuration also lies near one of the cataloged minima. The usual prescription for Smart Darting is to run the high temperature walker beforehand to obtain a set of quenched structures, $\{\mathbf{R}\}$, but for the 1-D surface investigated here we know the locations of all the minima and this is not necessary. We take the known positions of all the minima and form a set of "darts" (vector displacements between the minima). To do a Smart Darting jump move we then first check to see if both the current low-temperature configuration, $\mathbf{r}_l$, and the current quenched configuration, $\mathbf{r}_q$, lie within a small distance, $\varepsilon = 0.02$, of any minimum on the surface. If there exist minima i and j in the set $\{\mathbf{R}\}$ such that $|\mathbf{r}_l - \mathbf{R}_i| \leq \varepsilon$ and $|\mathbf{r}_q - \mathbf{R}_j| \leq \varepsilon$ $(i \neq j)$ then the jump occurs with the usual Boltzmann probability. If $\mathbf{r}_l$ or $\mathbf{r}_q$ lies outside all the cataloged ε-spheres then the current jump move is rejected. Note that for the Smart Darting simulations performed here $\mathbf{r}_q$ will always lie within ε of one of the minima. Thus the quenching

process is simply a way of selecting a particular dart move. The reason for selecting the dart moves in this way is that we wish to compare the methods to C-Walking, and part of the design of C-Walking is geared towards running the walkers in tandem.

For Parallel Tempering, the jump moves occur by randomly choosing a pair of walkers at neighboring temperatures, and attempting to swap the current configurations between the two temperatures as the next update in each other's Markov chain. The trial sampling distribution for drawing a particular configuration is $T(\cdot \rightarrow x_j) = \exp[-\beta_j V(x_j)]/Z_j$, where $V(x_j)$ is the potential energy of state x_j at the jth temperature, and Z_j is the configuration integral at that temperature. Swap moves between temperatures i and j are accepted with probability

$$\mathrm{acc}(x_i \rightarrow x_j) = \min[1, \exp(-(\beta_i - \beta_j)[V(x_j) - V(x_i)])] \quad (5)$$

which is similar to the J-Walking acceptance probability. However, unlike canonical J-Walking, Parallel Tempering *rigorously* satisfies detailed balance due to the exchange of configurations that couple the temperatures to form a continuous Markov chain. The drawback with having a large number of walkers at intermediate temperatures is that past a certain point there is minimal gain for the added expense of the intermediate walkers, as will be shown in the Results section.

C-Walking Method

C-Walking, like J-Walking and S-Walking, uses walkers at two temperatures. In contrast to the previous methods, C-Walking takes the high-temperature candidate configuration and begins cooling it by a statistical quenching process based on simulated annealing. The purpose of cooling the candidate high-temperature configuration is to attempt to bring its distribution more into "alignment" with the low-temperature one. Ideally one would cool the high-temperature configuration down to a temperature for which there is a significant increase in overlap; however, there are several problems that must be addressed in order to make this approach viable.

The first problem arises due to the finite length of any practical simulated annealing run. Even with an astutely chosen cooling schedule, successive configurations can become stuck in metastable minima, and one cannot be certain of arriving at the target temperature with properly distributed configurations. A method called annealed, or quenched, importance sampling developed by Neal[16] and Opps and Schofield[17] defines an importance sampler for the cooled configurations, allowing for meaningful averaging over quenched configurations. We briefly describe here our implementation of their method in the context of C-Walking, and refer the reader to the original sources for a more rigorous development.

Suppose we have decided to attempt a C-Walk jump move. We record the current high- and low-temperature configurations x_n and x_1, respectively, and then commence a cooling process from x_n, which generates a set of states at intermediate temperatures $\{x_{n-1}, x_{n-2}, \ldots\}$. For each state x_i we have a transition probability T_i that is assumed to produce the limiting distribution P_i. It is not a requirement for each T_i to produce ergodic sampling, and so we are free to use any of the usual MCMC methods. For our purposes it is convenient to use HMC updating.

The annealed importance sampling protocol begins by taking state x_n and updating it according to transition probability T_{n-1}. From this we produce a state with limiting distribution P_{n-1}, which we label x_{n-1}. x_{n-1} is then updated according to T_{n-2} to give a new state x_{n-2}, and so on. After j cooling steps we arrive at C-Walker configuration x_{n-j}, which we obtain as a result of the sequence $\{x_n, x_{n-1}, \ldots, x_{n-j+1}, x_{n-j}\}$.

In the end we wish to evaluate the transition probability $T(x_1 \rightarrow x_{n-j})$. In order to do this we need to first ascertain the appropriate weight for state x_{n-j}, that is, we need to calculate the probability associated with cooling from x_n to x_{n-j}. Following ref. 17 we assign a probability T_{cw} to the cooling process, given by

$$T_{\mathrm{cw}}(x_n \rightarrow x_{n-j}) \propto \frac{P_{n-j}(x_{n-j+1})}{P_{n-j+1}(x_{n-j+1})} \cdots \frac{P_{n-1}(x_n)}{P_n(x_n)} \quad (6)$$

This weighting essentially captures the "history" of steps in the cooling sequence. Trajectories that have become trapped in high-energy states receive appropriate low weighting. Note that this procedure obeys an equation of detailed balance, and produces properly weighted configurations that can be used to obtain equilibrium averages at the target temperature. The rate of convergence of averages produced in this way is dependent upon the variability of the weightings T_{cw}.

In order to maintain detailed balance, we need to determine a probability for the reverse process, that is, heating from x'_{n-j} to x'_n. For the reverse step we set $x'_{n-j} = x_1$ and determine the weight for the heating process of going from state x'_{n-j} to state x'_n, or equivalently, cooling from x'_n to x'_{n-j}

$$T_{\mathrm{cw}}(x'_n \rightarrow x'_{n-j}) \propto \frac{P_{n-j}(x'_{n-j+1})}{P_{n-j+1}(x'_{n-j+1})} \cdots \frac{P_{n-1}(x'_n)}{P_n(x'_n)} \quad (7)$$

A potential problem we would like to minimize is the necessity of performing an exorbitant number of cooling steps in order to generate trial configurations. We can reduce the number of cooling steps required by attempting to "jump out" of the cooling schedule early. Because we can assign a meaningful weight to the current C-Walker configuration at *any point* during the cooling process, we can define an acceptance criterion that maintains detailed balance for jumping out of the cooling cycle before it reaches the target temperature. Attempting to jump during the cooling process reduces the most costly aspect of the C-Walking approach, that is, the large number of cooling steps required to bring the trial configuration from the high temperature to a temperature (closer to the low temperature) for which there is a significant increase in jump-move acceptance. A successful jump during cooling terminates the cooling schedule and completes the C-Walk move.

Another way in which we lower the number of cooling steps is by utilizing "windows" of states during the attempt to jump to the current C-Walker configuration. The idea here is analogous to the orientational bias approach of Frenkel and Smit,[18] as well as the work of Neal,[19] and more recently Qin and Liu,[20] all of which demonstrate the potential for improved efficiency achieved in HMC by moving between windows of states. In C-Walking, collecting windows of states is an attempt to try to "wash-out" the fluctuations in potential energy present at the high temperature.

This is useful when attempting to exit the cooling cycle early because typically the jump will span a fairly large temperature difference. By using windows the number of necessary cooling steps is reduced by roughly 25%. On average the number of steps involved for C-Walking in this model is $\langle n_{\mathrm{cool}} \rangle \approx 70$ steps.

Putting all the pieces together, the exact C-Walking procedure is as follows. For a given HMC step, we decide whether or not to attempt a C-Walk jump by comparing a random deviate ξ to the jump probability. For $\xi < P_J$ a jump attempt is made, and we begin cooling the current high-temperature configuration while also continuing to propagate this same configuration at the high temperature. At each step in the cooling process we compare another random deviate to a probability for completing the C-Walk, that is, $\xi < P'_J$; a reasonable range of values for P'_J is roughly 3–20%. After some number of cooling steps j we attempt to complete the C-Walk move by halting the cooling process and sampling the current C-Walker configuration x_{n-j}.

At this point we have two configurations, x_1 and x_{n-j}. We set $x'_{n-j} = x_1$ and calculate $T_{\mathrm{cw}}(x'_{n-j} \rightarrow x'_n)$ by heating up x'_{n-j} using the annealed importance sampling protocol. Then starting from $x'_{n-j} = x'^{(1)}_{n-j}$ we generate a set $\{ x'^{(2)}_{n-j}, x'^{(3)}_{n-j}, \ldots, x'^{(n_w)}_{n-j} \}$ at the C-Walker temperature, and we also generate a corresponding set starting from x_{n-j} to give $\{ x^{(1)}_{n-j}, x^{(2)}_{n-j}, \ldots, x^{(n_w)}_{n-j} \}$; here n_w is a simulation parameter specifying the number of states to collect in the windows. From the window of states $\{x^{(1)}_{n-j}, x^{(2)}_{n-j}, \ldots, x^{(n_w)}_{n-j}\}$ we select a configuration $x^{(i)}_{n-j}$ with probability $\exp[-\beta_{n-j} V(x^{(i)}_{n-j})]$. The jump $x_1 \rightarrow x^{(i)}_{n-j}$ is now accepted with probability

$$\mathrm{acc}(x_1 \rightarrow x^{(i)}_{n-j}) = \min\left[1, \frac{P_{n-j}(x_1) P_1(x^{(i)}_{n-j}) T_{\mathrm{cw}}(x_n \rightarrow x_{n-j})}{P_1(x_1) P_{n-j}(x^{(i)}_{n-j}) T_{\mathrm{cw}}(x'_n \rightarrow x'_{n-j})} \frac{W(x_{n-j})}{W(x_1)}\right] \quad (8)$$

where

$$W(x_1) = \exp[-\beta_{n-j} V(x_1)] + \sum_{k=2}^{n_w} \exp[-\beta_{n-j} V(x'^{(k)}_1)] \quad (9a)$$

and

$$W(x_{n-j}) = \sum_{k=1}^{n_w} \exp[-\beta_{n-j} V(x^{(k)}_{n-j})] \quad (9b)$$

If the jump is accepted, the low-temperature configuration is updated, and regular HMC updates are resumed; otherwise we continue the cooling from the current C-Walker configuration.

Note that for the full duration of the cooling procedure we continue to propagate the high-temperature walker. When the C-Walk move is completed we stop propagation of the high-temperature walker and record its final configuration, which is stored away until the next C-Walk attempt. If we reach the low temperature without any successful jump we simply save the last high-temperature configuration and return to propagating the low-temperature walker by itself via HMC. Because the jump probability is typically set to be fairly low, we end up propagating two walkers in tandem for only a fraction of the total run time; however, the actual number of total steps for the high-temperature walker is essentially the same as in J-Walking and S-Walking. The advantage here is that the high-temperature steps performed are concentrated solely on breaking up the correlations at the high temperature in between jumps.

Finally, a few words should be said about the HMC protocol. In HMC one moves from an initial point in phase space (**x, p**) by resampling the momenta **p** from a Maxwell distribution, and then propagating the system forward in time for n_{MD} time steps to arrive at a final point (**x′, p′**). The new configuration, **x′**, is then used as the candidate configuration for the next state in the Markov chain. Because energy conservation is not required, one is free to choose MD time steps that are substantially larger than normal. The size of the time step, and the number of MD steps per HMC cycle, are parameters that are dependent on the particular system being investigated. Given initial configuration x with Hamiltonian $H(x,p)$, and candidate configuration y with Hamiltonian $H(y, p')$, the Metropolis criterion for $x \rightarrow y$ is

$$\mathrm{acc}(x \rightarrow y) = \min[1, \exp(-\beta[H(y, p') - H(x, p)])] \quad (10)$$

Provided the integration algorithm used in the MD steps is time reversible and area conserving, this procedure produces a Boltzmann distributed Markov chain. In our implementation we use the velocity version[21] of the classic Verlet algorithm for the integration steps in the MD updates. We use a time step for the MD steps that is four times larger than the necessary time step to perform constant energy molecular dynamics. n_{MD} is set equal to five for all HMC cycles. It is worth pointing out here that a nice benefit to using HMC with C-Walking is that it has been demonstrated to be an effective method for performing simulated annealing, as it results in more efficient thermalization of the degrees of freedom in the system per HMC cycle.[22]

Results

A plot of the potential energy function is shown in Figure 1, along with the exact probability distribution functions at the reduced temperatures of the high- and low-temperature walkers, $T^* = 3.0$ and 0.1, respectively. The exact distribution functions are calculated explicitly from the expression $\rho_{\mathrm{exact}}(x) = \exp(-\beta\ V(x))/Z$, where Z is the configuration integral.

Being able to calculate the exact distribution allows us to construct two excellent measures for the sampling accuracy and efficiency of each method. During every simulation run we keep a running tab of the developing normalized probability distribution as calculated by simply binning the position of the particle as it moves across the surface. By comparing the probability distribution $\rho(x)$ to the exact distribution $\rho_{\mathrm{exact}}(x)$ we are able to gauge the sensitivity and accuracy of each method.

Shown in Figure 2 is $\rho(x)$ averaged over multiple independent runs for the four methods. For the most part all methods appear to do equally well except for S-Walking. The discrepancy becomes apparent on inspection of the two smallest peaks, where S-Walking shows an expected over-weighting of configurations near the minimum.

Table 2. Comparison of Ergodic Sampling Methods for $P_J = 3\%$.

	J-walking	S-walking	Smart darting	Parallel tempering	C-walking
$\langle x \rangle$	2.7	2.7	2.7	2.8	2.8
$\langle V \rangle$	−3.6	−3.6	−3.6	−3.6	−3.6
r_{jump}	5%	29%	17%	30%	54%

Exact averages for position and potential energy are $\langle x \rangle = 2.8$ and $\langle V \rangle = -3.6$, respectively. r_{jump} is the percent of successful jump moves for the low-temperature walker. The parallel tempering results are for the case of one intermediate walker.

To investigate the rate at which configurations are sampled, we evaluate the time scale of the approach of $\rho(x, t)$ to the exact distribution. As in ref. 14 we calculate the following quantity during each simulation run:

$$\chi^2(t) = \int_0^L dx[\rho(x, t) - \rho_{\text{exact}}(x)]^2 \tag{12}$$

As a trajectory moves over the surface, it samples configurations that, if properly weighted, produce a distribution function $\rho(x; t)$ with asymptotic limit $\rho_{\text{exact}}(x)$. $\chi(t)$ provides a measure of the rate at which a method's sampling approaches the exact distribution. Obviously to capture the full distribution on this surface a walker will require some minimum number of jumps. For an ergodic system, $\chi(t)$ eventually decays to zero.

Shown in Figure 3 are plots of $\chi(t)$ for all methods with jump probability $P_J = 3\%$. The time axis used is the number of MD steps executed by the low-temperature walker. This choice of time scale allows for easy comparison between the different methods. The jump acceptance probabilities, r_{jump}, for the low-temperature walker are shown in Table II for the different methods. The Parallel Tempering data shown are for one intermediate walker. It can be seen that J-Walking has a much slower initial decay than S-Walking, Smart Darting, or C-Walking. The rate of initial decay in $\chi(t)$ for Parallel Tempering is similar to J-Walking; however, at the longer times it can be seen that Parallel Tempering appears to have decayed further and has a steeper slope. The C-Walking data decays faster and to lower values than those of any of the other methods.

The inferior sampling of J-Walking is due to the much lower acceptance probability for jump moves between the widely separated high and low temperatures. At short times, S-Walking shows a relaxation on par with Smart Darting and C-Walking due to the efficiency with which it performs hopping between basins. The high plateau in $\chi(t)$ for S-Walking at longer times shows the consequence of not weighting the basins appropriately at the target temperature.

While the sampling in C-Walking clearly outperforms Smart Darting in its initial rate of decay, the decay at longer times is close between the two methods. The performance of Smart Darting is very sensitive on having a complete set of darts for the surface at hand. On this 1-D surface the results of Smart Darting are altered if only a single dart is removed from the set. One could argue that in practice if Smart Darting were to miss a particular minimum, then any of the other methods could also be expected to miss that minimum. However, this points to another advantage of C-Walking that is tied to running the two walkers in tandem. In Smart Darting, if the high-temperature walker is run insufficiently long such that an important region of configuration space is neglected, one is forced to stop the low-temperature simulation, rerun the high-temperature trajectory to find more states from which to construct the missing darts, and then restart the low-temperature run. For C-Walking one can simply continue to run the simulation until sufficient convergence is achieved, as the high-temperature propagation is done concurrently.

The rate of sampling in Parallel Tempering is inferior to all the methods considered here except J-Walking. Most likely this is due to the increased diffusiveness of the exploration of parameter space induced by the presence of intermediate walkers. We illustrate this in Figure 4 by showing $\chi(t)$ for an increasing number of walkers. It can be seen that increasing the number of walkers does not result in an enhanced rate of convergence for $\chi(t)$, despite the sharp increase in jump acceptances in going from 30% (one walker) to 82% (nine walkers).

Conclusions

We have presented a new method, called C-Walking, for overcoming quasi-ergodicity problems with simulations on rough energy surfaces. C-Walking uses one high-temperature walker run in tandem with the low-temperature walker, and offers a practical advantage in that it provides the low-temperature walker with uncorrelated trial moves sampled from the high temperature walker. By defining a transition probability T_{cw} that captures the correct weighting of the cooling stage, as well as the corresponding transition probability defined similarly for the reverse heating process, C-Walking correctly weights the exchange between *any* intermediate (cooled) configuration and the configuration at the low temperature. This allows us to define an arbitrarily large internal jump rate for exiting the cooling cycle early, thereby saving significant computational expense. We further reduce the number cooling steps by jumping between windows of states to improve jump-acceptance probabilities.

The advantage of C-Walking over J-Walking is that it is able to achieve larger jump acceptance while continuing to implement only two walkers. Thus it realizes a greater degree of basin-hopping and therefore has superior ergodic sampling compared to J-Walking. Unlike S-Walking, C-Walking satisfies detailed bal-

ance, and so weights corresponding basins appropriately for bringing configurations from the high temperature to the low temperature. In addition, C-Walking also realizes much more efficient ergodic sampling than either J-Walking or S-Walking as measured by the ergodicity factor $\chi(t)$.

In comparison to Smart Darting, C-Walking realizes faster decay in $\chi(t)$ at short times as well as convergence to a lower value at longer times. Smart Darting is a way to correct for detailed balance in S-Walking, and has a place as a potentially powerful technique for overcoming quasi-ergodicity. The drawback with Smart Darting has to do with the fact that its effectiveness as a method depends upon having a representative set of dart vectors for the given surface of interest. For systems with large numbers of minima, this may become problematic as the management of a comprehensive catalogue of all pertinent darts becomes a daunting task. Exceedingly large sets of darts in systems with many degrees of freedom might require special techniques for searching through the dart matrix.

The reason C-Walking outperforms Parallel Tempering, which does not have problems with detailed balance, has to do with the increase in the diffusiveness of the exploration of parameter space as the number of walkers increases. If we add a single additional walker at a temperature intermediate between the high and low temperatures, the number of successful configuration swaps between the low and intermediate temperatures and the high and intermediate temperatures will increase. This makes sense as the walkers at adjacent temperatures are now closer in temperature and should possess greater overlap between their distributions. It might seem that this situation should enhance the rate of exploration of configuration space, but in actuality this is not always the case. The intermediate walker retards the movement of high-temperature configurations to the low temperature. In effect the increased jump acceptance achieved with a larger number of walkers is off-set by an increase in the time required for configurations from the high-temperature walker to reach the low temperature. The lag induced by mediated swapping through the intermediate walkers becomes larger as the number of walkers increases. In the limit of walkers at a continuum of temperatures the swap acceptance probability would be 1, but exploration through parameter space would become a random walk.

In light of this, another important advantage of C-Walking compared to Parallel Tempering is that it is able to realize very closely spaced intervals in temperature while not losing the aforementioned benefit of having only two walkers. This stems from the fact that C-Walking cools slowly, and to a certain degree executes walkers at a continuum of temperatures, which more sensitively and quickly find the optimal overlap with the distribution at the target temperature.

Additionally, unlike most of the other methods discussed here where one must "fine-tune" multiple parameters before production runs can occur, in C-Walking there is much less need to fine-tune the protocol for the system of interest. It is naturally built into the C-Walking method, as it explores just enough of a cooling trajectory to find the best overlap. Note that some tinkering may be necessary with the cooling schedule for any chosen system, although a conservative cooling schedule will always be an unambiguous choice. Unlike C-walking, many of the methods suffer from correlations between temperature walkers when run in tandem. Although this can be improved by running the high-temperature walker as an independent simulation and saving configurations, which are then sampled randomly during the low-temperature simulation, this will certainly be untenable for any realistic system of interest.

In regards to the computational cost, as noted previously both J-Walking and Parallel Tempering have essentially a constant computational cost with increasing jump rate. On the other hand, S-Walking, Smart Darting, and C-Walking have costs that scale with jump rate when the walkers are run in tandem. For instance, at $P_J = 1\%$ C-Walking is roughly 30% more expensive than S-Walking, whereas for $P_J = 3\%$ the computational cost is roughly four times that of S-Walking. However, when combined with the fact that C-Walking converges at least 10 times more rapidly than S-Walking at $P_J = 3\%$, it is clear that C-Walking provides a much more efficient sampling for the cost.

As a final comment we note that although the C-walking method does very well on a 1-D surface, the real test of course comes from simulation of systems with much more realistic levels of complexity. Work on implementing C-Walking in simulations with greater complexity is currently underway, and initial investigations indicate that it is indeed viable for more complicated systems.

References

1. Metropolis, N.; Rosenbluth, A.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. J Chem Phys 1953, 21, 1087.
2. Northup, S. H.; McCammon, J. A. Biopolymers 1980, 19, 1001.
3. Pangali, C.; Rao, M.; Berne, B. J. Chem Phys Lett 1978, 55, 413.
4. Rossky, P. J.; Doll, J. D.; Friedman, H. L. J Chem Phys 1978, 69, 4628.
5. Duane, S.; Kennedy, A. D.; Pendleton, B. J.; Roweth, D. Phys Lett B 1987, 195, 216.
6. Berne, B. J.; Straub, J. E. Curr Opin Struct Biol 1997, 7, 181.
7. Frantz, D. D.; Freeman, D. L.; Doll, J. D. J Chem Phys 1990, 93, 2769.
8. Schofield, J. Private communication.
9. Matro, A.; Freeman, D. L.; Topper, R. Q. J Chem Phys 1996, 104, 8690.
10. Hukushima, K.; Nemoto, K. J Phys Soc Jpn 1996, 65, 1604.
11. Geyer, C. J.; Thompson, E. A. J Am Stat Assoc 1995, 90, 909.
12. Lyubartsev, A. P.; Martsinovski, A. A.; Shevkunov, S. V.; Vorontsov-Velyaminov, P. N. J Chem Phys 1992, 96, 1776.
13. Marinari, E.; Parisi, G. Europhys Lett 1992, 19, 451.
14. Zhou, R.; Berne, B. J. J Chem Phys 1997, 107, 9185.
15. Andricioaei, I.; Straub, J. E.; Voter, A. F. J Chem Phys 2001, 114, 6994.
16. Neal, R. M. Stat Comp 2001, 11, 125.
17. Opps, S. B.; Schofield, J. Phys Rev E 2001, 63, 056701.
18. Frenkel, D.; Smit, B. Understanding Molecular Simulations; Academic Press: San Diego, 2002; p 323.
19. Neal, R. M. J Comp Phys 1994, 111, 194.
20. Qin, Z. S.; Liu, J. S. J Comp Phys 2001, 172, 827.
21. Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. J Chem Phys 1982, 76, 637.
22. Salazar, R.; Toral, R. J Stat Phys 1997, 89, 1047.