Smart walking: A new method for Boltzmann sampling of protein conformations

Ruhong Zhou and B. J. Berne
Department of Chemistry and the Center for Biomolecular Simulation, Columbia University, New York, New York 10027

(Received 3 June 1997; accepted 29 August 1997)

A new Monte Carlo algorithm is presented for the efficient sampling of protein conformation space called the Smart-Walking (S-Walking) method. The method is implemented using a hybrid Monte Carlo protocol. The S-Walking method is closely related to the J-Walking method proposed by Frantz *et al.* (J. Chem. Phys. **93**, 2769, 1990). Like the J-Walking method, the S-Walking method runs two walkers, one at the temperature of interest, the other at a higher temperature which more efficiently generates ergodic distributions. Instead of sampling from the Boltzmann distribution of the higher temperature walker as in J-Walking, S-Walking first approximately minimizes the structures being jumped into, and then uses the relaxed structures as the trial moves at the low temperature. By jumping into a relaxed structure, or a local minimum, the jump acceptance ratio increases dramatically, which makes the protein system easily undergo barrier-crossing events from one basin to another, thus greatly improving the ergodicity of the sampling. The method approximately preserves detailed balance provided the time between jumps is large enough to allow effective sampling of conformations in each local basin. © *1997 American Institute of Physics.* [S0021-9606(97)51845-9]

I. INTRODUCTION

Computer simulations of protein structures and dynamics have been of great interest in the past decade. It is now widely believed that the complexity of the potential energy landscape results in the rich dynamical behavior of proteins.[1–4] The rugged energy surface arises from the heterogeneous nature of proteins because of the presence of many energy scales. For example, the barriers are due to at least two classes of interactions: first, local barriers separate stable torsion angle states; second, barriers arise from close encounters of atoms among the sidechains. The equilibrium and dynamical properties of proteins are thought to be determined by this temperature-independent multidimensional potential energy hypersurface consisting of many local minima and barriers.

When molecular dynamics (MD) or Monte Carlo (MC) simulations are used to determine the conformational equilibrium of proteins or other biomolecules, the underlying assumption is that the average over the simulation trajectory (the trajectory or time average) is equal to the average over all possible states of the system (the statistical or phase space average). This is often called the ''ergodic hypothesis.'' However, due to the finiteness of the simulation time, the trajectory average often differs from the phase space average; and the trajectory averages obtained from different starting configurations may have very different values. In this case, the sampling scheme is said to be ''non-ergodic'' or ''quasi-ergodic.''

How to avoid this ''quasi-ergodicity'' problem in real protein systems still remains a great challenge in computer simulations. Quasi-ergodicity appears whenever the time scale of MD or MC simulations is shorter than an important relaxation time scale in proteins, or when the protein system is trapped in a local potential basin. Unfortunately, the time scales for functionally important motions, such as protein folding, are often longer than accessible MD or MC runs by today's computing resource.[4,5] Therefore, enhanced sampling algorithms which improve the rate of barrier crossings are required for the search of conformation space.

Enhanced sampling algorithms should be designed in such a way as to allow protein systems to have a significant probability for making long-range moves or simply visiting barrier regions. It is not practical for conventional MC methods to make nonlocal collective moves because of the prohibitively low acceptance ratio. It is also not practical for normal MD methods to efficiently sample barrier crossings due to the very small time steps required by energy conservation. One efficient method for generating collective moves is the hybrid Monte Carlo (HMC) method invented by Duane and Kennedy.[6] In this method, one starts with a configuration of the system and samples the momenta of the particles from a Maxwell distribution. Molecular dynamics is then used to move the whole system for a time Δt, and finally one accepts or rejects the move by a Metropolis criterion based on $\exp(-\beta H)$ where H is the Hamiltonian of the whole system. A number of authors have further elaborated the HMC method, and have applied it to many systems, such as spin glasses and polymers.[7–11]

The HMC method also gives rise to one practical problem. Since the momenta are constantly refreshed, the accompanying dynamics is similar to Smoluchowski dynamics and thus gives a spatial diffusion process superimposed on the inertial dynamics. This added spatial diffusion can lead to smaller rates for barrier crossing.[3] One way to improve this is to couple it with the J-Walking method proposed by Frantz, Freeman and Doll.[12] Unlike the conventional MC

method which samples a small move and accepts or rejects the move based on Metropolis criterion, in the simplest incarnation, the J-Walking method runs two walkers, one at the temperature of interest, the other, called the J-Walker, at a higher temperature. The lower temperature walker can occasionally jump into the J-Walker's configurations, and the jump is then accepted or rejected based on a Metropolis criterion such that detailed balance is preserved for the lower temperature walker. The occasional sampling from the J-Walker's Boltzmann distribution, which samples a larger conformation space due to its higher temperature, allows the lower temperature walker to easily move form one basin to another and thus speeds up the conformation space sampling. A similar method, called the exchange Monte Carlo method, was introduced by Hukushima *et al.*[13] for spin glasses. In practice, it has been found that the use of only a single high-temperature J-Walker (two-stage) is insufficient to treat quasi-ergodicity in physically realistic systems (even small clusters).[14] A multi-stage J-Walking must then be used. Obviously, the CPU and memory cost will increase linearly with the number of stages. The computational cost for simulating large protein systems may then be enormous.

There are many other methods proposed to enhance the conformation space sampling, such as anti-force-bias MC by Cao and Berne,[15] multicanonical MC by Berg and Neuhaus,[16] the cluster move method by Swendsen and Wang,[17] the fluctuating potential method by Liu and Berne,[18] the Monte Carlo-minimization approach by Li and Scheraga[19,20] and the mixed Monte Carlo/stochastic dynamics method by Guarneri and Still.[21,22] These methods have shown some success in model systems, small clusters or Lennard-Jones systems. The Monte Carlo-minimization method generates a MC move for a system, followed by a minimization of the structure, which then accepts or rejects the minimized structure by the Metropolis criterion.[19,20] The method is very successful in a global minimum search for a pentpeptide, Met-enkephalin; however, the distribution it generated is no longer canonical and the system might still be trapped in a ''deep'' local minimum (not global minimum) at low temperatures. The mixed MC/stochastic dynamics method,[21,22] which alternates small moves generated by stochastic dynamics with large MC moves of certain torsion angles, has been quite successful for chain molecules in continuum solvent, but should face difficulties in explicit solvents. Very recently, Andricioaei and Straub also proposed an enhanced sampling method based on the Tsallis statistical distribution[23] and applied it to atomic clusters. However, the problem of sampling protein conformation space efficiently still remains unsolved.

In this paper, we present a new Monte Carlo algorithm called the Smart-Walking (S-Walking) method for efficient sampling of protein systems. The method is implemented with a HMC protocol, although it could also be implemented with an ordinary Monte Carlo. The beauty of this new method is that it often requires only two stages for effective sampling compared with J-Walking which often requires multi-stages. Thus, S-Walking will often be computationally much less costly than J-Walking. In the following, we compare S-Walking (which requires only two walkers) with two-stage J-Walking and find S-Walking samples conformation space much more efficiently. Since a multi-stage J-Walking procedure will increase CPU cost linearly with the number of stages, we focus only on the comparison of two-stage J-Walking with S-Walking.

This paper is organized as follows: several ergodic measures for measuring the ergodicity of the sampling are described in Section II, followed by three sections on methodology, HMC (Section III), J-Walking (Section IV) and S-Walking (Section V). Section VI gives some application results on a one dimensional random potential surface and two protein systems, a pentpeptide Met-enkephalin and a protein melittin (PDB file 2mlt, 431 atoms). Section VII contains the conclusion. The basic approach introduced in this paper can easily be used in conjunction with other schemes to enhance the rate of conformation sampling.

II. GENERALIZED ERGODIC MEASURES

In order to compare the new S-Walking algorithm with other methods such as J-Walking, it is essential to define a measure of the efficiency of a method and to determine if a given sampling method is ergodic.

If the true normalized probability distribution, $\rho_{exact}(\Gamma)=Z^{-1}\exp(-\beta V(\Gamma))$, of the conformational states is known, and if $\rho(\Gamma,t)$ is the normalized distribution found from simulation after a time t, one can monitor the quantity[15]

$$\chi^2(t)=\int_{-\infty}^{+\infty} d\Gamma[\rho(\Gamma,t)-\rho_{exact}(\Gamma)]^2 \tag{1}$$

as a function of run length. Here, Γ denotes the conformational space, and the integral is over all conformational states. The root mean square deviation should decay to zero if the sampling method is ergodic and if the run length exceeds the mixing time of the sampling algorithm. Then the rate of decay of $\chi(t)$ is a measure of how efficient a simulation method is. For one-dimensional energy landscapes, one knows $\rho_{exact}(x)$, so that $\chi(t)$ offers a simple measure of sampling efficiency as was shown in Ref. 15. We use $\chi(t)$ on a simple one-dimensional random potential later in Section VI. For multi-dimensional systems with rugged energy landscapes, however, the exact normalized distribution is usually not known. Then one must search for other measures of sampling efficiency.

Thirumalai *et al.*[24] have introduced an approximate method for determining whether or not a sampling algorithm is ergodic that also produces a measure of the mixing time. The underlying idea is that for any system in equilibrium, independent trajectories over an ergodic system must be self-averaging. Take a property such as the pair distance between a particular pair of sites in a protein. If the protein system dynamics is ergodic, the average value of the pair distance over two independent trajectories α and β must be equal. This condition is of course not sufficient for ergodicy of the system, but it is necessary.

Several ergodic measures, such as a force measure (based on vector force of an atom), and an energy measure (based on the non-bonded energy of an atom) among others, have been proposed by Thirumalai *et al.*[24] and Straub *et al.*[4] In general, for a quantity g of the jth atom or jth pair, the average over time t for a particular trajectory is defined as

$$\overline{g_j^{\alpha}}(t)=\frac{1}{t}\int_0^t ds\ g_j^{\alpha}(s), \tag{2}$$

where α indicates the average calculated over α trajectory. For two independent trajectories α and β starting from independent initial configurations, we define the mean-square difference between averages of trajectory α and β as the metric

$$d_g(t)=\frac{1}{N}\sum_{j=1}^{N} |\overline{g_j^{\alpha}}(t)-\overline{g_j^{\beta}}(t)|^2. \tag{3}$$

This metric $d_g(t)$ is often called the ergodic measure.[24,4] There are many possible choices for the quantity g. Several different measures will be discussed in this paper, such as the force magnitude metric, the potential energy metric, the 1-4 pair distance metric, and the all-pair distance metric. As we will see later, the force metric, energy metric and 1,4-pair distance metric will measure more about local sampling; however, the all-pair distance metric will give more information about the long-range sampling.

III. HYBRID MONTE CARLO METHOD

Conventional Monte Carlo simulations are generally carried out by means of single-particle moves. Updating more than one particle at a time for large protein systems typically results in prohibitively low average acceptance ratios. On the other hand, MD simulations can perform global moves, however, the MD scheme is prone to errors and instabilities due to finite time step sizes. Thus only very small time steps (δt = 0.5 fs for proteins) can be used for normal MD simulations.

The hybrid Monte Carlo (HMC) method proposed by Duane *et al.*[6] combines the ease of global update by MD and the Metropolis criterion of MC. It is unlike conventional MC methods, however, because it involves global updates of positions of all atoms based on an accept/reject decision for the whole configuration. It is also unlike the normal MD scheme because there are no discretization errors due to finite time step size. In general, the time step in HMC could be 2–3 times larger than the normal MD time step for proteins while keeping the method exact and with sufficiently high acceptance ratio. The beauty of the HMC method is that it is actually an *exact* MC method with global updates.

In the HMC method, one starts with an initial state of the system $(\mathbf{r},\mathbf{p})$, and resamples momenta $\mathbf{p}$ from a Maxwell distribution. Molecular dynamics is used to move the whole system for a time $\Delta t = n_{\mathrm{MD}}\delta t$, where δt is the time step of the MD simulation, and n_{MD} is the number of MD steps in one MC cycle. Because Δt is chosen sufficiently large that the total energy is not conserved, one then accepts or rejects the move based on the Metropolis criterion,

$$p=\min[1,\exp(-\beta\Delta H)], \tag{4}$$

where p is the acceptance probability, and H is the Hamiltonian of the system. This whole process is called one HMC cycle. The HMC cycle is repeated over and over until a long trajectory is generated. Thus, in summary, the HMC algorithm can be described as follows:

(1) Starting from an initial position in phase space $(\mathbf{r},\mathbf{p})$, resample the momenta $\mathbf{p}$ from a Maxwell distribution.
(2) Perform n_{MD} steps of a constant energy MD simulation, with a time step δt, to generate a new configuration $(\mathbf{r}_{new}, \mathbf{p}_{new})$.
(3) Accept or reject this new configuration based on the Metropolis criterion on the total Hamiltonian, Eq. (4), then go back to step (1).

It can be shown that HMC leads to the canonical probability distribution, provided that the MD algorithm is time reversible and symplectic.[6] Since our multiple time step algorithm, the so-called reversible Reference System Propagator Algorithm (r-RESPA)[25–30] is time reversible and symplectic, it is very useful in combination with HMC for large protein systems.

In practice, the HMC method should be optimized for different systems by fine tuning the time step δt and number of MD steps n_{MD} used in one MC cycle. For many systems,[8–11,31] HMC can sample the phase space very efficiently; however, it is still not efficient enough for large protein systems, as we will see in the Results section. One way to improve it is to couple the HMC method with the Jump-Walking (J-Walking) or Smart-Walking (S-Walking) techniques discussed in the following sections. In the results section, we will see that the combination of HMC with S-Walking greatly speeds up the conformational sampling of proteins over J-Walking and pure HMC.

IV. JUMP WALKING

The Jump-Walking (J-Walking) method was first proposed by Frantz, Freeman and Doll[12] in MC studies of atomic clusters. It has been shown that J-Walking is very powerful in reducing the quasi-ergodic behavior of model systems and small atomic clusters.[12,32,14]

The J-Walking technique can be described as follows. In the usual MC method, one samples a small move and accepts or rejects the move according to the Metropolis criterion at the temperature of interest,

$$p=\min[1,q(x'|x)], \tag{5}$$

where

$$q(x'|x)=\frac{T(x|x')\rho(x')}{T(x'|x)\rho(x)}, \tag{6}$$

$\rho(x)=Z^{-1}\exp[-\beta V(x)]$ is the Boltzmann distribution with Z the partition function, and $T(x'|x)$ is the trial sampling distribution. The trial sampling distribution adopted in MC methods is usually a uniform distribution over a step size Δ,

$$T(x'|x)=\begin{cases}\dfrac{1}{\Delta}, & |x'-x|<\dfrac{\Delta}{2}\\ 0, & otherwise\end{cases}. \tag{7}$$

This gives the well-known Metropolis probability function $q(x'|x)$,

$$q(x'|x)=\exp\{-\beta[V(x')-V(x)]\}. \tag{8}$$

In the J-Walking method, this normal sampling is infrequently punctuated by sampling from a higher temperature distribution for the same system. Since a higher temperature MC simulation can involve larger attempted moves and more frequent barrier crossings, this allows the system to access more conformational states according to the high temperature Boltzmann distribution. Then, the lower temperature walker attempts occasional jumps to the conformation states of the high temperature walker, thus enhancing the barrier crossing. The trial sampling distribution for these occasional jumps is the Boltzmann distribution at the higher temperature

$$T_J(x'|x)=Z^{-1}\exp[-\beta_J V(x')], \tag{9}$$

which, from Eq. (6), gives the acceptance probability function $q(x'|x)$ as

$$q_J(x'|x)=\exp\{-(\beta-\beta_J)[V(x')-V(x)]\}. \tag{10}$$

In the limit $\beta_J\to 0$, $q_J(x'|x)$ reduces to the standard Metropolis acceptance probability in Eq. (8). This is reasonable because the high temperature distribution broadens as β_J decreases to zero, and the J-Walker method then essentially reduces to simple jumping with a large step size Δ_J in this limit. In the limit of $\beta_J\to\beta$, $q_J(x'|x)\to 1$ since the low temperature walker is now effectively sampling from its own distribution.

It should be noted that the Hamiltonian is used in the HMC method and the potential energy is used in the J-Walking method (S-Walking too). This is permissible since both methods generate the same canonical Boltzmann distribution.

The easiest way to implement the J-Walking method is to run two walkers in tandem. However, this will result in large correlations between two walkers and thus large systematic errors. Another way to implement J-Walking method is to run the J-Walker first and generate an external file of the configurations at the high temperature, then run the low temperature walker and infrequently sample from this external file randomly. Doll *et al.*[12,32,14] found that the second method is more efficient and generates less systematic error than the first method. However, the first method, running two walkers in tandem, is more convenient for parallel machines. Thus, it might be appropriate to use tandem walkers in parallel machines and external files in serial machines. Since there are a huge number of conformations for real proteins, enormous disk storage may be needed for the external file in the second method, thus the two methods are combined in our implementation (we use serial machines here). The modified J-Walking algorithm based on a hybrid Monte Carlo (HMC) protocol is then as follows:

(1) Starting from some initial configuration, run a J-Walker at a high enough temperature T_J using the HMC method for N MC cycles, and generate M conformations uniformly (one every N/M HMC cycles). Store these M conformations in an external file. Memorize the last configuration of the J-Walker for future use.
(2) Starting from the initial configuration, run the low temperature walker using the HMC method for N MC cycles. During the HMC run, randomly jump to the J-Walker conformations stored in the external file, when a randomly generated number $\xi<P_J$ (P_J is the jump probability). The jump is accepted or rejected according to Eq. (10). At the end, store the last configuration of the low temperature walker.
(3) Assign the last configurations of J-Walker and low temperature walker as their new initial configurations, respectively, then go back to step (1) and repeat the process until the simulation converges or a predefined number of steps is exceeded.

The starting configuration of the two walkers is the same in our implementation, but that is not necessary. This J-Walking method preserves detailed balance for the low temperature walker, and the occasional jumps to the J-Walker configurations allow the low temperature walker to move from one potential energy basin to another, thus reducing the CPU time required to sample conformation space. This approach appears to work very well for model systems and small atomic clusters[12,32,14] of great interest to determine how well the J-Walking method works for large real protein systems, which was the initial goal of this research project. Unfortunately, the normal two-stage J-Walking method (only two temperatures) has very limited success in protein systems (see the Results section).

V. SMART WALKING

As mentioned above, in order for the J-Walking method to work well for large protein systems, it might need a large number of stages of walkers and thus a large amount of parallel processors. The reason is obvious because the accessible temperature of the J-Walker needs to be very close to that of the low temperature walker to have a reasonable jump acceptance ratio. If too high a temperature is used for the J-Walker, configurations are generated in which some groups may overlap and the van der Waals (vdW) interactions are sufficiently large that most of these will be rejected by the low temperature walker. In other words, if the difference between the temperatures of the low temperature walker and J-Walker is large, the average potential energy of the J-Walker will be much higher than that of the low temperature walker in terms of kT, and the jump success ratio will be extremely low. For example, the jump success ratio for the pent-peptide Met-enkaphalin jumping from 300 K to 400 K is less than 5%. It will be even less for larger proteins.

Thus, a multi-stage jumping process instead of a two-stage jumping scheme is necessary for real systems. For example, a three-stage jump walking scheme ($T_1>T_2>T_3$) can be described as follows: a low temperature walker at

temperature T_3 can occasionally jump to a intermediate temperature walker at T_2, and the intermediate walker at T_2 occasionally jumps to the high temperature walker at T_1, provided T_1 is higher enough for the MC simulation to be ergodic. This process can be extended to n-stage jumping processes. However, because CPU time and memory requirement increases linearly with the number of stages, this many stage process is very resource consuming. Matro, Freeman and Topper[14] proposed a parallel jump-walking scheme, which uses as many as 90 processors in parallel for atomic clusters $(NH_4Cl)_n$, $n=3-10$. Hukushima and Nemoto[13] proposed a similar method called the exchange Monte Carlo Method, which used 32 stages in the jumping process for three-dimensional spin glass systems with lattice size $L=6-16$.

However, jumping directly into a high temperature structure is not the only way to use the conformational space information from the J-Walker. Instead, the structure can be first relaxed before being jumped into. Approximate minimization with a steepest descent method (or conjugate gradient method) will generate structures close to the local minimum,

$$\{x'\} \overset{relax}{\Rightarrow} \{x''\}, \tag{11}$$

where $\{x'\}$ is the configuration of the J-Walker before minimization, $\{x''\}$ is the configuration after minimization. These relaxed configurations will significantly decrease the potential energy, and thus increase the jump success ratio dramatically. However, since the relaxation process is a non-thermal process, the minimized structures $\{x''\}$ no longer satisfy the Boltzmann distribution at high temperature. Fortunately, the high temperature Boltzmann distribution need not to be satisfied in order to use the minimized structures. Instead, we regard a minimized structure as one of the possible trial moves at low temperature and use the normal acceptance probability function,

$$q(x''|x)=\exp\{-\beta[V(x'')-V(x)]\}. \tag{12}$$

Unlike the J-Walking acceptance probability in Eq. (10), this scheme, which we call Smart-Walking, or S-Walking, will dramatically increase the jump success ratio from one basin to another. It also enables the system to explore more phase space and undergo more efficient barrier crossings. This S-walking method avoids the linear increase of CPU time and memory usage required by the multiple-stage J-Walking method, because it is not necessary to use multiple stages for most systems, even though it would be very easy to implement a multi-stage S-Walking procedure. S-Walking preserves detailed balance approximately provided the time between S-jumps is much longer than the time required by the low temperature walker to explore its local basin effectively (a further discussion follows in the section on Results). This new S-Walking algorithm only requires a simple modification of the J-Walking algorithm.

(1) Starting from some initial configuration, run a J-Walker at high enough temperature using the HMC method for N HMC cycles. Generate M conformations uniformly (one every N/M HMC cycles), and approximately *minimize* each of them to their corresponding local minima using steepest descent method. Store these minimized M conformations in an external file. Memorize the last configuration of the J-Walker for future use.
(2) Starting from the initial configuration, run the low temperature walker using the HMC method for N MC cycles. During the HMC run, randomly attempt a jump to one of the minimized conformations stored in the external file, when a random number $\xi<P_S$ (P_S is the jump probability in S-Walking). The jump is accepted or rejected according to Eq. (12). Store the last configuration of the low temperature walker.
(3) Assign the last configurations of the J-Walker and the low temperature walker as their new initial configurations, respectively, then go back to step (1) and repeat the process until the simulation converges or a predefined number of steps is exceeded.

VI. RESULTS AND DISCUSSION

We have applied the S-Walking algorithm to several systems and compared it with other methods, such as the HMC and J-Walking algorithms. Constant energy MD and constant temperature MD, are also included for comparison. A one-dimensional rugged potential surface is tested first, for which the exact normalized probability distribution is known. Then, two protein systems, a pentpeptide Met-enkephalin and a protein Melittin (PDB file 2mlt), are tested.

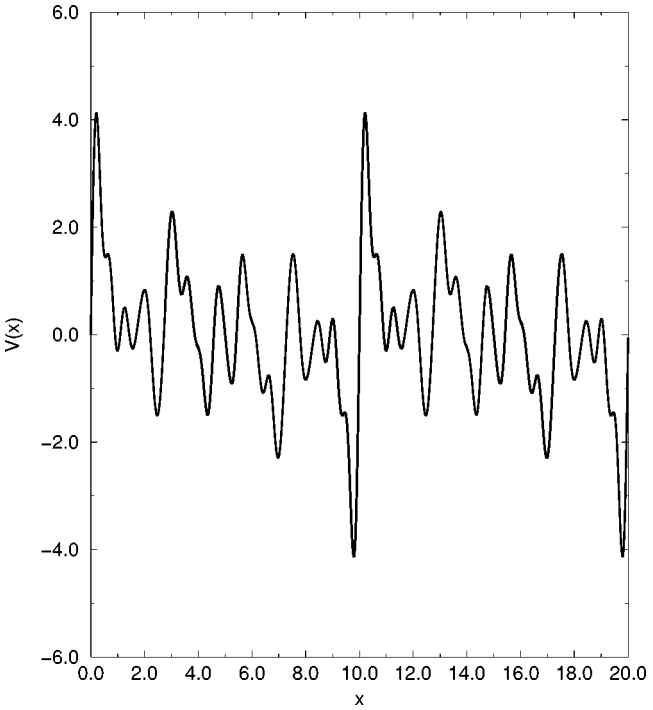

FIG. 1. Diagram of a one-dimensional random potential energy surface from a sum of Fourier sine waves with periodic boundary condition.

A. One-dimensional random potential surface

We begin with a simple one-dimensional (1D) model system. A Fourier sum of sine waves is used to generate a rugged potential surface,

$$V(x)=\sum_n C_n \sin\left(\frac{n\pi x}{L}\right). \tag{13}$$

The potential $V(x)$ is periodic in the x direction with L the size of the primary cell, and C_n the coefficient of the nth component. By changing the coefficients C_n and the number of terms n, various 1D potential surfaces can be generated. In this study, $L=10$, $n=20$ and a randomly generated coefficient set $\{C_n\}$ are used. The generated 1D potential surface, as shown in Figure 1, has a global minimum at $x=9.7$ and various local minima between $x=0$ and $x=10$.

The units of the 1D potential energy are arbitrary, and other parameters are set to be reduced units, such as the mass $m=1.0$, and the Boltzmann constant $k=1.0$. Thus the reduced temperature, T^* will be 2KE, where KE is the kinetic energy of the particle with the same units as the potential energy.

Five different methods are tested: constant energy MD (CEMD); constant temperature MD (CTMD); hybrid Monte Carlo (HMC); the Jump Walking method based on HMC (J-Walking) and the Smart-Walking method based on HMC (S-Walking). In the CTMD method, a simple velocity scaling scheme proposed by Berendson is used.[33]

The time step δt used in CEMD is determined by energy conservation, ΔE, which is defined as[27]

$$\Delta E=\frac{1}{N}\sum_{i=1}^{N}\left|\frac{E_{initial}-E_i}{E_{initial}}\right|, \tag{14}$$

where E_i is the total energy at step i, $E_{initial}$ is the initial energy, and N is the total number of time steps. A requirement of $\log(\Delta E)\leqslant -3.0$ is used to determine the time step in CEMD for different temperatures. The CTMD method shares the same time step δt as the CEMD for the same temperature. In the HMC method, two parameters are used: the MD time step δt and the number of MD steps n_{MD} in each MC cycle. In general, the time step adopted in HMC can be much larger than that of CEMD. For the simple 1D random potential surface, the time step of HMC can be as large as 10 times that of CEMD with the resulting acceptance ratio still higher than 50%. However, as we will see later in the real protein systems discussed in this paper, the MD time step in HMC can be only a factor of 2–3 times larger than that of the CEMD (the time step in HMC will decrease as the size of protein systems increase). For a reasonable comparison, we used a factor of 4 in the time step for this 1D system with n_{MD} equal to 5. Both the J-Walking and S-Walking methods are based on an HMC protocol, with the same jumping probability of $P_J=P_S=3\%$ (the jumping probability of J-Walking P_J will be increased to 10% for protein systems in order to increase the jump success rate). In the J-Walking method, a high temperature walk is performed first using HMC at $T^*=3.0$ (high enough for HMC to sample all states in a reasonable time), and an external file of these high temperature configurations is generated uniformly (every 20 MC cycles or 100 MD steps). A low temperature walk is then generated, which occasionally (3% of the moves) jumps into the configurations stored in the external file. The S-Walking method follows the same procedure as the J-Walking method, except that a relaxation process (implemented with the steepest descent method) is added before storing the high temperature configurations to the external file. Since the required disk storage of this 1D system is moderate, we can generate the high temperature walk first and store all resulting configurations (10 000 structures) in one file.

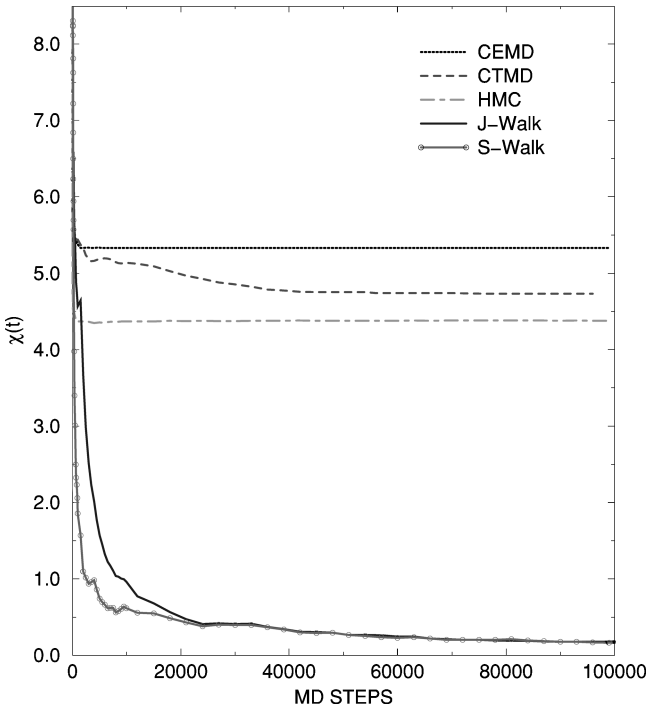

FIG. 2. A plot of χ, as defined in Eq. (1), versus the number of MD steps for the 1D rugged potential surface at temperature $T^*=0.1$ using various methods. It is shown that constant energy MD (CEMD) and constant temperature MD (CTMD) methods cannot cross barriers efficiently. The hybrid Monte Carlo (HMC) method improves the ergodic sampling to some extent at this temperature. Both the J-Walking (J-Walk) and S-Walking (S-Walk) methods work very well, with S-Walking working a little better. However, as we will see later, the S-Walking method will be much more efficient than the J-Walking method in real protein systems.

Since we know the exact form of the 1D rugged potential energy surface, it is easy to calculate the *exact* probability distribution function, $\rho_{exact}(x)$. Thus, $\chi(t)$, as defined in Eq. (1) can be used as a criterion for ergodic measurement of the sampling. If the sampling is ergodic, $\chi(t)$ should decay to zero. The rate of decay is a measure of the convergence of sampling, which indicates how fast the walker accesses all possible states.

Figure 2 shows the decay of $\chi(t)$ as a function of the number of MD steps for the various methods at temperature $T^*=0.1$, starting from an initial position of $x=6.9$ (a local minimum). Here we used the number of MD steps, not the MD real time t, as our time axis, because the time steps used

in normal MD methods and HMC methods are different. Another reason for using MD steps is because they reflect the CPU time. It should be noted that the CPU time for J-Walking is about twice that of regular MD because there are two walkers; and the CPU cost of S-Walking is about 3 times that of regular MD because of the extra cost of minimization procedures. The CPU time of HMC is comparable with normal MD methods for the same number of MD steps. It is clear from the figure that for both CEMD and CTMD, $\chi(t)$ decays to large plateau values, which means that these normal MD methods cannot cross barriers efficiently, resulting in non-ergodic sampling. Thermal averages based on these simulation methods depend on the initial conformational states. In order for the CEMD method to be ergodic in this 1D system, the temperature has to be raised to about $T^*=13.0$ (kinetic energy larger than ~ 6.5). In contrast, HMC is ergodic at any temperature, but at low temperatures, barrier crossings will be so infrequent that the conformational sampling is in a practical sense ''non-ergodic.'' In this 1D system, when $T^*<1.5$, we observe this ''non-ergodic'' behavior in less than 500 000 MD moves or equivalently 100 000 HMC moves ($n_{\mathrm{MD}}=5$). For $T^*>1.5$, HMC samples the rugged landscape very efficiently. Thus, for very low temperatures, like $T^*=0.1$, there is little difference between HMC and CEMD/CTMD as shown in Figure 2, but HMC is markably superior to CEMD/CTMD at $T^*>1.5$, well before CEMD/CTMD becomes ''ergodic.'' The reason why we select a very low temperature here is to clearly show the difference between HMC and the J-Walking method, and also the difference between the J-Walking and S-Walking methods. Both the J-Walking and S-Walking methods generate a significant decay for $\chi(t)$ at $T^*=0.1$, indicating that both methods can access all possible states in this rugged 1D potential surface even at very low temperatures. Of course, as we can see from the figure, the S-Walking method is still 3–4 times faster in the initial decay than the J-Walking method. Accounting for the fact that the S-Walking method requires 50% more CPU time, it is a factor of 2–3 times more efficient than the J-Walking method for this 1D system. When the temperature is increased, the difference between HMC, J-Walking, and S-Walking methods for this 1D model systems becomes smaller, because HMC already crosses barriers efficiently at higher temperatures. However, as we will see in the next section, the difference between these three methods will become more significant in multi-dimensional systems. The HMC and the two-stage J-Walking methods will then have very limited success in sampling protein systems, whereas the S-Walking method (two-stage) still works very well.

As shown in Frantz *et al.*'s paper, the J-Walking method preserves detailed balance. The S-Walking method only preserves detailed balance approximately, and generates a Boltzmann distribution provided the time interval between jumps is much longer than the energy relaxation time. Figure 3 shows the spatial distribution function, $\rho(x)$, calculated from the S-Walking method (after 100 000 HMC steps, or 500 000 MD steps) and from exact numerical evaluation for two different temperatures, $T^*=0.1$ and $T^*=1.0$. The distribution

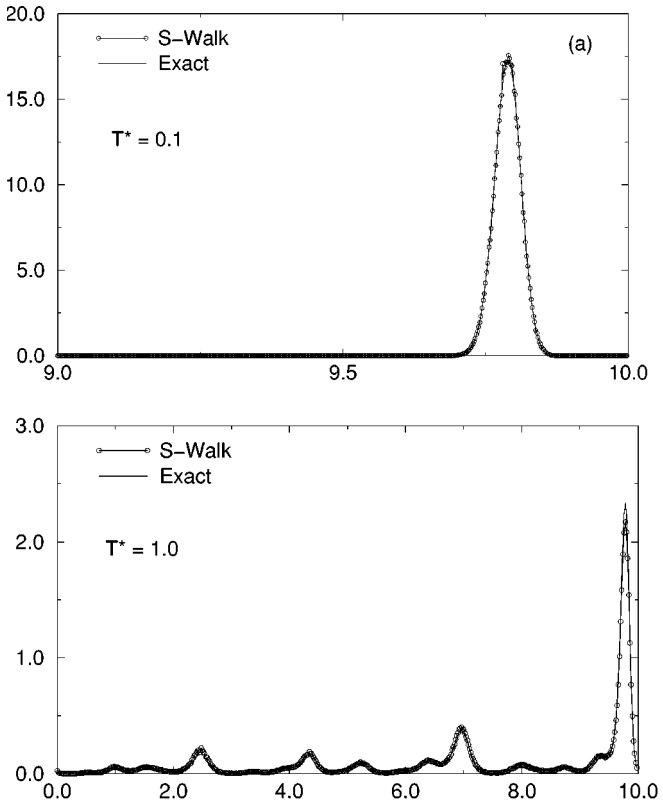

FIG. 3. A comparison of the probability distribution function, $\rho(x)$, obtained from S-Walking simulation and from the exact evaluation, for two different temperatures: (a) $T^*=0.1$, (b) $T^*=1.0$. The results show that the S-Walking method with $P_S=3\%$ generates correct Boltzmann distributions for both temperatures.

function calculated from the S-walking method agrees very well with the *exact* distribution function for both temperatures, indicating that the S-Walking method generates the Boltzmann distribution. However, it should be pointed out that if very large jumping probability P_S is used, the S-Walking method will spend a lot of time just sampling the local minima, not the thermal distribution of configurations around each of these minima, resulting in an over-weighting in these local minima configurations. To reduce this possible over-weighting of the local minima, the jump frequency should be low enough so that the time interval between jumps will be much larger than the energy relaxation time. Fortunately, the jump success ratio for each jump is very high, so the overall barrier crossing probability is still much higher than the two-stage J-Walking method.

Figure 4 shows the decay of $\chi(t)$ with time at temperature $T^*=0.1$ using various S-Walking jumping probability P_S. As expected from the above discussion, $\chi(t)$ does not decay to zero for P_S larger than 10% because of the over-weighting of the local minima. Clearly, if P_S is made too small, the relaxed structures will be chosen so rarely that very long runs will be required to obtain effective sampling of the correct distribution. This is shown in Figure 4 for $P_S=0.3\%$. Thus, it behooves the simulator to determine the optimal value of P_S for various systems. We note that meth-

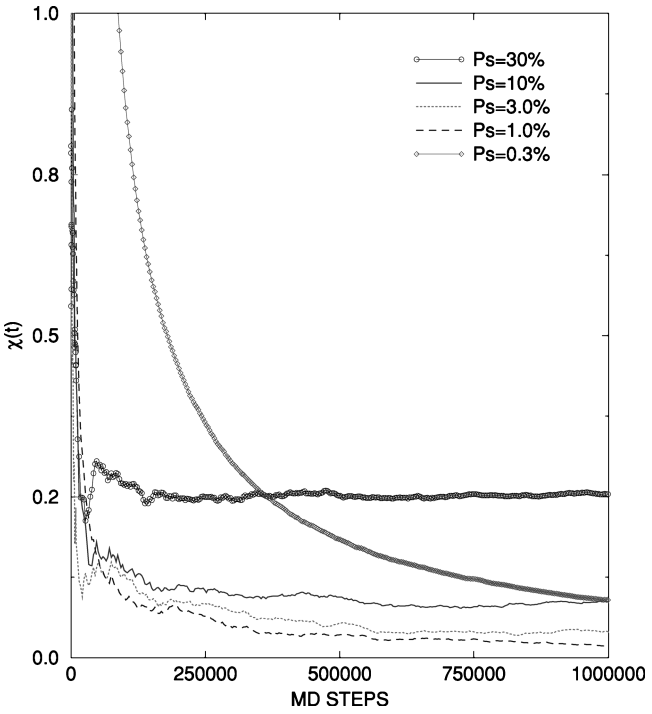

FIG. 4. The decay of $\chi(t)$ with time, i.e., the number of MD steps in HMC, for the 1D rugged potential surface at temperature $T^*=0.1$ using various jumping probability P_S in S-Walking. The results show that the optimal S-Walking jump probability is about 1–10% for this 1D system.

ods based on the maximization of the rate of information entropy production might be useful for this optimization.[34] The optimal value of P_S is found to be 1–10% for this 1D random potential surface. In this paper a $P_S=3\%$ is used for both 1D random potential and protein systems. The optimal value of P_S may depend on the relative curvature of the barrier regions and local minima in potential hypersurfaces.

In some extreme cases, such as a broad potential well with a very sharp hole in the center (a ''needle-point''), the minimization process in S-Walking will locate the ''needle point'' as the trial move for the low temperature walker, resulting in a serious over-weighting of this point. Thus, a very small P_S should be used for this kind of potential surface. Alternatively, another approach can also be used to reduce this local minima over-weighting in which after the energy minimization the system is thermalized at the low temperature using HMC before Metropolis acceptance. This alternative approach (which is not implemented in this paper) will have an additional CPU cost for the thermalization around local minima at the low temperature, but may use a larger jumping probability P_S. Which of the two methods will be more efficient for various systems is currently under investigation.

B. Application to two protein systems

As we have seen above, both the two-stage J-Walking and S-Walking methods work very well for one-dimensional rugged potential surfaces. In fact, HMC also works well for not-too-low temperatures. Protein systems have multi-dimensional rugged conformational hyper surfaces. Will these methods still work for large protein systems?

Two protein systems are studied: one is an oligopeptide, Met-enkephalin (residue sequence Tyr-Gly-Gly-Phe-Met), and the other is a protein melittin from honey bee (PDB file 2mlt, 26 residues and 431 atoms). An all-atom model is used for both protein systems, and the AMBER force field[35] is used in all calculations. All simulations are performed by incorporating the new methodologies into the biomolecular simulation package, IMPACT.[36]

However, unlike the 1D model system, it is impossible to calculate the *exact* spatial probability distribution function $\rho_{exact}(\Gamma,t)$ for protein systems. Thus, $\chi(t)$ is no longer a useful measure of sampling for these real systems. We adopt ergodic measures, $d(t)$, introduced by Thirumalai *et al.*,[24,5] to measure the convergence of the conformational space sampling approximately. However, in order to make these ergodic measures meaningful, one must run two (at least) simulations starting from two independent initial conformations. It could be difficult in practice to determine which states are independent. If the two initial structures α and β are dependent on each other, or they are too close in phase space (for example, if they are separated by only a small barrier), the results would be misleading. Here, we follow Straub *et al.*'s[4] method for preparing the two initial independent configurations, α and β. The configuration α is chosen from a 50 ps MD equilibration at 300 K using velocity scaling. The configuration β is obtained by gradually heating up the pentpeptide and melittin to 1000 K (25 ps MD), then quenching down to 300 K, followed by another 50 ps equilibration at 300 K. All the simulations are then started from these two configurations, α and β. In the J-Walking and S-Walking simulations, a total number of $M=500$ structures are generated (and stored in an external file) from the high temperature walker every $N/M=20$ HMC cycles (total $N=20\times500=100\,000$ HMC cycles) with or without minimization (100 steps of minimization by steepest descent). The low temperature walker is then followed for the same number of HMC cycles (100 000 HMC cycles) with occasional jumps to the structures in the external file. The above process is repeated until the simulation converges or a predefined number of steps is exceeded. The number of HMC steps ($N/M=20$) separating these stored structures should be large enough to eliminate the correlation between them; and the total number of structures M should be as large as possible consistent with the disk space available and small enough to guarantee reasonable direct access time.

Three ergodic measures are used in this study: (a) a force metric $df(t)$ for the magnitude of force on each atom; (b) a 1,4-pair distance metric $d14(t)$ for 1,4-pair distances; and (c) a pair distance metric $dij(t)$ for all pairs (for large proteins, we can pick a fixed number of pairs uniformly to save disk space),

$$df(t)=\frac{1}{N}\sum_{k=1}^{N}|\overline{f_k^{\alpha}}(t)-\overline{f_k^{\beta}}(t)|^2,$$

$$d14(t)=\frac{1}{N'}\sum_{k=1}^{N'}|\overline{r14_k^{\alpha}}(t)-\overline{r14_k^{\beta}}(t)|^2, \tag{15}$$

$$dij(t)=\frac{1}{N''}\sum_{k=1}^{N''}|\overline{rij_k^{\alpha}}(t)-\overline{rij_k^{\beta}}(t)|^2.$$

where N,N',N'' are number of atoms, number of 1,4 pairs, and number of all pairs, respectively, and $r14, rij$ are 1,4 and i,j pair distances. The force metric $df(t)$ measures the magnitude of total force on each atom. However, the force is weighted by local interactions, thus it is often a measure of the local sampling. On the other hand, the pair distance metric $dij(t)$ measures all pair distances, but its value is affected more by long-range than short-range pair distances, so it weights the long-range overall similarities between two structures. The 1,4-pair distance metric only measures local sampling as is obvious from its definition. As we will see later, the all-pair distance metric $dij(t)$ is a very sensitive and demanding measure for the ergodic sampling. The vector force metric, $d\vec{f}(t)$, for vector forces on each atom, as used by Thirumalai and Straub *et al.*,[24,4] has also been tried for these protein systems. However, the time average of the vector forces decays to zero very quickly due to the cancellation of forces pointing in different directions. Therefore, it becomes difficult to discern the differences between various methods using the vector force metric. Meanwhile, the energy metric is similar to the force magnitude metric. So, we do not include the vector force metric or energy metric in this study.

Figures 5(a)–(c) shows these three ergodic measures, $df(t)$, $d14(t)$ and $dij(t)$, for the pentpeptide Met-enkephalin obtained from various methods. As expected, both the CEMD and CTMD methods (time step $\delta t=0.5$ fs) give a non-decaying plateau in the three ergodic measures, because normal MD methods do not generate barrier crossings efficiently. The two pentpeptide configurations are trapped in two different local minima, and the time average of the pair distance or force magnitude for each configuration are not able to change with time after equilibrium at the local basin. The CTMD method is slightly better than the CEMD method in general, as shown in Figure 5, because the fluctuation of the total energy in CTMD helps it access more of the conformation space. The HMC method improves the sampling to some extent because a larger time step can be used. The optimal HMC parameters for this pentpeptide are found to be δt = 1.0–1.5 fs (2–3 times larger than normal MD) and n_{MD} = 5–10. In this study, δt = 1.5 fs and $n_{\mathrm{MD}}=5$ are used. The results show that the HMC method does indeed enhance the ergodic sampling for this pentpeptide. The force metric $df(t)$ decays to zero after 10 000 MD steps, and the pair distance metrics $d14(t)$ and $dij(t)$ both decay to a smaller value compared to normal MD methods. However, $d14(t)$ and $dij(t)$ do not decay to zero, instead, $d14(t)$ shows a plateau around 0.46, and $dij(t)$ a plateau around 0.65 even after 500 000 MD steps. This means that the pair distance metrics are much more sensitive and also demanding than the commonly used force or energy metrics.[24,4] Similar results are also found in the J-Walking and S-Walking methods, as well as in the larger protein system, melittin. The reason for this is that different structures may have the same forces or energies. However, requiring all pair distances to be self-averaging is a very demanding requirement, thus $dij(t)$ is a very strong ergodic measure, even though it may still not be sufficient.

The J-Walking method uses the same optimized parameters as HMC. As the J-Walker's temperature T_J increases, the jump success ratio decreases dramatically. This is because the average potential energy of the pentpeptide at a high temperature is much higher than the potential energies acceptable at the lower temperature of 300 K. Thus, the jump acceptance ratio, Eq. (10) decreases to almost zero when T_J is increased to about 500 K. To maintain a reasonable jump acceptance ratio, for example 5–10%, we must choose the temperature T_J to be about 350–400 K for this pentpeptide. However, the J-Walker at these temperatures (350–400 K) cannot efficiently sample the conformational states within a reasonable CPU time, for example 500 000 MD steps (real proteins will denature at these temperatures though). To make the high temperature walker efficiently sample states in a reasonable CPU time (500 000 MD steps), the temperature T_J must be raised above 1000 K. Thus, for the J-Walking method to work properly, a multi-stage walker must be used (for example, jumping from 300 to 350 K, and 350 to 450 K, etc.). The temperature interval ΔT between two consecutive stages need not be taken equal, instead these intervals can be optimized by requiring the same jump success ratio between stages. In this study, only the normal two-stage J-Walking scheme is used for comparison. A J-Walker at 400 K is used with a higher jump trial probability of $P_J=10\%$ (the jump probability for S-Walking is still taken as $P_S=3\%$. The results in Figures 5(a)–(c) show that in the J-Walking method, the decay rate of our ergodic measures improves compared to HMC, but not as dramatically as in the 1D system. Also, shown in Figure 5(b), the J-Walking method has a larger plateau value for the $d14(t)$ measure than that of the HMC method. This is one example where the ergodic measures are only approximate measures. Sometimes, such measures can be misleading (see the discussion at the end of this section).

Unlike the J-Walking method, the S-Walking method allows the use of a much higher temperature T_S for the high-temperature walker. Because the high temperature structures are relaxed, the average potential energy of the relaxed structure is comparable or even lower than that of the low temperature walker at 300 K. The results show that ergodic measures, $df(t)$, $d14(t)$ and $dij(t)$, all decay to zero much faster than the J-Walking method. The most demanding metric $dij(t)$ decays to zero after 400 000 MD steps, indicating that the quasi-ergodic behavior in the other methods is greatly reduced in the S-Walking method.

A larger protein melittin, with 431 atoms and more than 1200 degrees of freedom, is also tested. Similar results are shown in Figures 6(a)–(c). For this system, a time step of 0.5

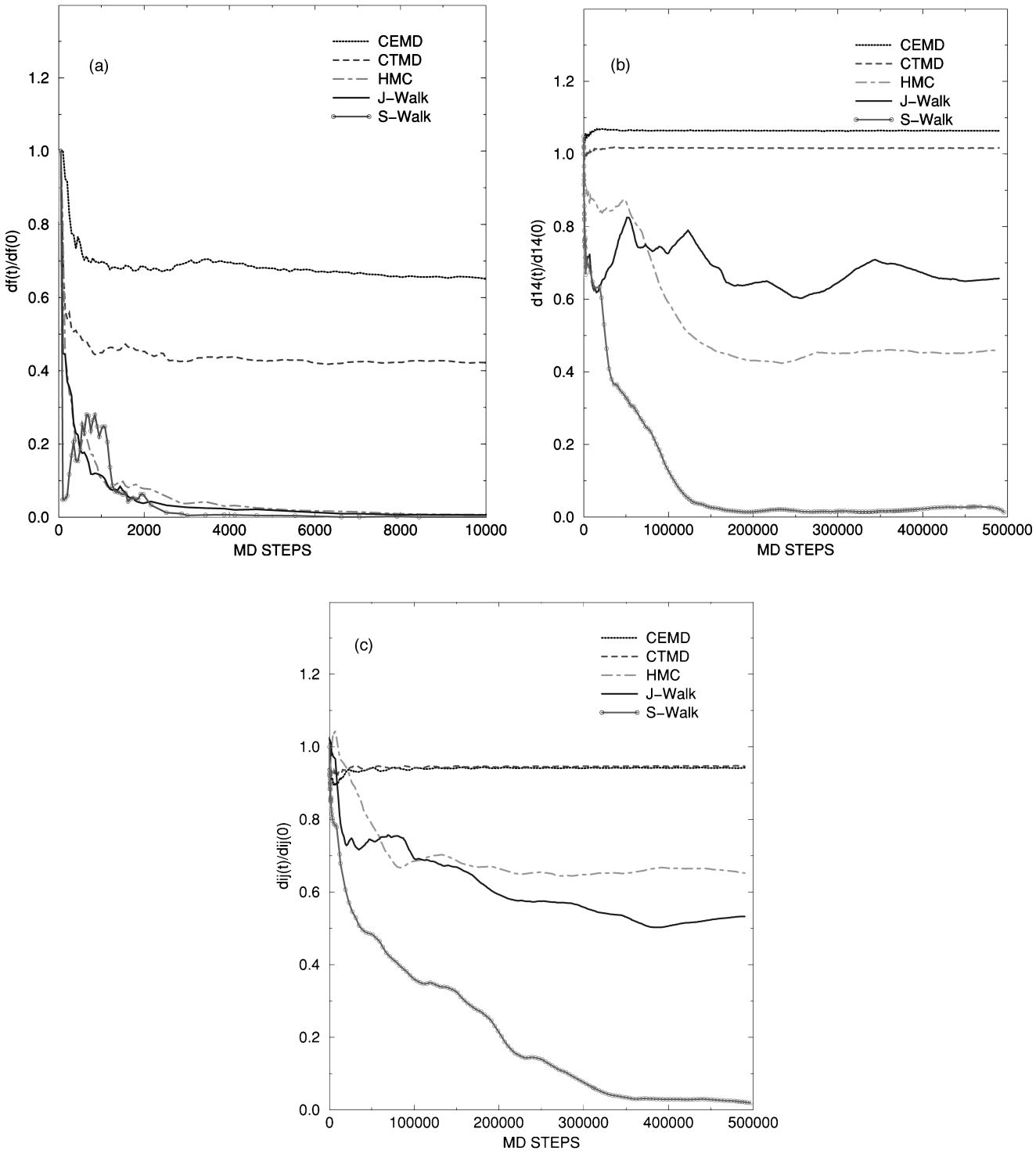

FIG. 5. Decay of the three ergodic measures with the number of MD steps for the pentpeptide Met-enkephalin using various sampling methods: (a) force magnitude metric $df(t)$; (b) 1,4-pair distance metric $d14(t)$; (c) all-pair distance metric $dij(t)$. It should be noted that the CPU time for J-Walking is about twice that of regular MD or HMC, and the CPU time for S-Walking is about three times that of regular MD or HMC. See the text for details.

fs is also used in the CEMD and CTMD methods, and a time step of 1.0 fs and $n_{\mathrm{MD}}=5$ is used in the HMC method. The J-Walker temperature T_J is reduced further to 350 K to achieve a reasonable jump success ratio. As can be seen from the figure, for this larger system, the difference in the results obtained from HMC and J-Walking is smaller. Both the HMC and J-Walking methods have limited success for this larger protein system. On the other hand, the S-Walking method (with $T_S = 1000$ K) still works very well. Of course, we do not see the pair distance metrics decay to zero within

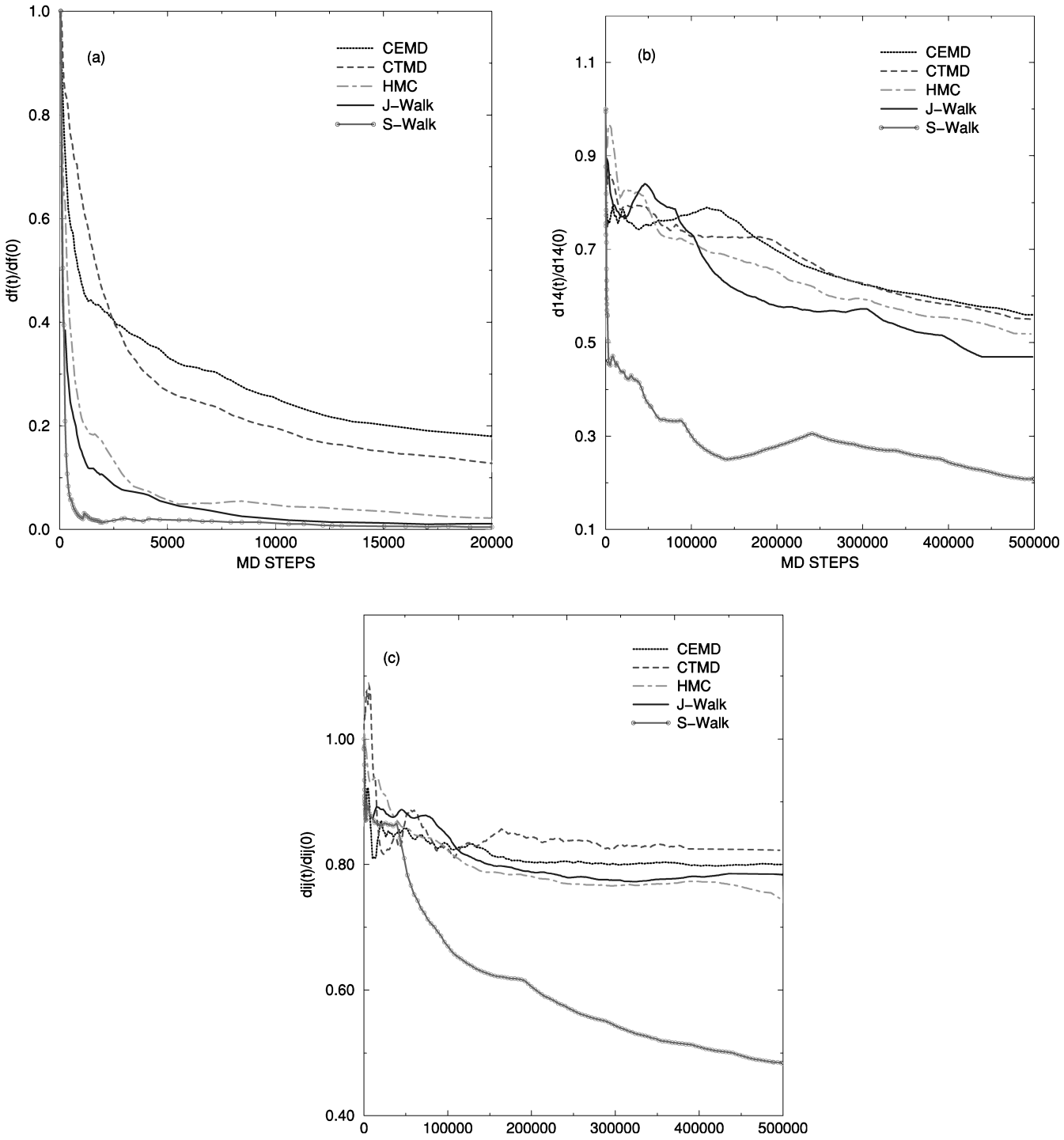

FIG. 6. Decay of the three ergodic measures with the number of MD steps for the protein melittin (PDB file 2mlt) using various sampling methods: (a) force magnitude metric $df(t)$; (b) 1,4-pair distance metric $d14(t)$, (c) all-pair distance metric $dij(t)$. It should be noted that the CPU time for J-Walking is about twice that of regular MD or HMC, and the CPU time for S-Walking is about three times that of regular MD or HMC. See the text for details.

500 000 MD steps, because it takes more time for the method to sample the much larger conformation space of this larger protein.

Finally, it should be noted that the requirement of having two initial independent structures is very crucial for these ergodic measures to be meaningful. As mentioned above, if the two structures are dependent or too close in phase space, the less efficient methods might give better ergodic measures because the two simulations are essentially sampling the same local basin. On the other hand, more efficient methods may enable the system to cross barriers more easily, and thus access other phase regions, which may result in a temporary

increase, not decrease, in ergodic measures with time. Here again, the rate of information entropy production might provide a useful measure.[34]

VII. CONCLUSION

In this paper, a new algorithm, the Smart-Walking method coupled with hybrid Monte Carlo, has been presented for efficient sampling of conformational phase space. The S-Walking method is developed from the J-Walking method proposed by Frantz, Freeman and Doll.[12] It has been shown that the S-Walking (two-stage) method will greatly reduce the quasi-ergodic sampling for multi-dimensional rugged potential systems, such as protein systems; while the J-Walking (two-stage) method faces difficulties in large protein systems due to the enormous rejection rate in trial jumps.

Both the S-Walking and J-Walking method run two walkers (or more in the parallel version) at two different temperatures (or multi-level temperatures); one is at the temperature of interest and the other is at a higher temperature which can generate ergodic distributions. Instead of jumping to a Boltzmann distribution of the higher temperature walker as in J-Walking, S-Walking first minimizes the high temperature configurations, and then uses these relaxed structures as the trial moves for the low temperature. By jumping to relaxed structures, or local minima, the jump acceptance ratio increases dramatically, which makes it possible for proteins to undergo barrier crossings from one basin to another, thus greatly improving the ergodicity of the sampling. The method approximately preserves detailed balance and generates a Boltzmann distribution provided the time interval between jumps is much longer than the energy relaxation time.

The S-Walking method has been applied to a one-dimensional rugged potential surface, and two protein systems, Met-enkephalin and melittin. It has been shown that the S-Walking method can sample conformational phase space more efficiently than other methods, including the J-Walking method. In general, a normal two-stage J-Walking method increases the CPU time by a factor of 2, and the S-Walking method increases the CPU cost by a factor of 3. However, for the J-Walking method to have a reasonable jump acceptance ratio for real protein systems, hierarchical multi-stage walkers are often required and thus result in a linear increase of the CPU time and memory requirement with the number of stages. On the other hand, the S-Walking method only requires two-stage walkers for most systems because the relaxed structures have much lower potential energy than those unrelaxed structures in J-Walking.

In the paper, we used HMC as the underlying sampling technique for J-Walking and S-Walking. Of course, one could also use single particle Monte Carlo moves instead. It is also very convenient to couple the S-Walking method with the sate-of-the-art algorithms RESPA/FMM[27] and RESPA/PFMM[30] for even larger protein systems in vacuum or in solvent.

ACKNOWLEDGMENTS

This work was supported by a grant from the National Institutes of Health (ROI GM43340) and from the NIH Division of Research Resources (SP41RR06892). One of the authors (R.Z.) wishes to thank Yong-Han Lee and Dr. Emilio Gallicchio for some useful discussions. R.Z. also wishes to thank the Pittsburgh Supercomputing Center for a gift of computer time as part of the DEC Computational Chemistry Award he received from the American Chemical Society, 1995.

[1] J. A. McCammon and S. C. Harvey, *Dynamics of Proteins and Nucleic Acids* (Cambridge University Press, Cambridge, 1987).
[2] K. M. Merz Jr. and S. M. Le Grand, *The Protein Folding Problem and Tertiary Structure Prediction* (Birkhäuser, Boston, 1994).
[3] B. J. Berne and J. E. Straub, Curr. Opinion Struct. Biol., **7**, 181 (1997).
[4] J. E. Straub, A. B. Rashkin, and D. Thirumalai, J. Am. Chem. Soc. **116**, 2049 (1994).
[5] J. E. Straub and D. Thirumalai, Proteins, **15**, 360 (1993).
[6] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[7] P. B. Markenzie, Phys. Lett. B **226**, 369 (1989).
[8] M. E. Tuckerman, B. J. Berne, G. J. Martyna, and M. L. Klein, J. Chem. Phys. **99**, 2796 (1993).
[9] A. Irback, J. Chem. Phys. **101**, 1661 (1994).
[10] B. M. Forrest and U. W. Suter, J. Chem. Phys. **101**, 2616 (1994).
[11] D. G. Gromov and J. J. dePablo, J. Chem. Phys. **103**, 8247 (1995).
[12] D. D. Frantz, D. L. Freeman, and J. D. Doll, J. Chem. Phys. **93**, 2769 (1990).
[13] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[14] A. Matro, D. L. Freeman, and R. Q. Topper, J. Chem. Phys. **104**, 8690 (1996).
[15] J. Cao and B. J. Berne, J. Chem. Phys. **92**, 1980 (1990).
[16] B. A. Berg and T. Neuhaus, Phys. Lett. B **267**, 249 (1991).
[17] R. H. Swendsen and J. S. Wang, Phys. Rev. Lett. **58**, 86 (1987).
[18] Z. Liu and B. J. Berne, J. Chem. Phys. **99**, 6071 (1993).
[19] Z. Li and H. A. Scheraga, Proc. Natl. Acad. Sci. USA **84**, 6611 (1987).
[20] Z. Li and H. A. Scheraga, J. Mol. Struct. **179**, 333 (1988).
[21] F. Guarneri and W. C. Still, J. Comput. Chem. **11**, 1302 (1994).
[22] H. Senderowitz, F. Guarneri, and W. C. Still, J. Am. Chem. Soc. **117**, 8211 (1995).
[23] I. Andricioaei and J. E. Straub, J. Chem. Phys. (submitted).
[24] D. Thirumalai, R. D. Mountain, and T. R. Kirkpatrick, Phys. Rev. A **39**, 3563 (1989).
[25] M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).
[26] D. D. Humphreys, R. A. Friesner, and B. J. Berne, J. Phys. Chem. **98**, 6885 (1994).
[27] R. Zhou and B. J. Berne, J. Chem. Phys. **103**, 9444 (1995).
[28] R. Zhou, S. J. Stuart, and B. J. Berne, J. Chem. Phys. **105**, 235 (1996).
[29] S. J. Stuart, R. Zhou, and B. J. Berne, J. Chem. Phys. **105**, 1426 (1996).
[30] F. Figueirido, R. Zhou, R. Levy, and B. J. Berne, J. Chem. Phys. **106**, 9835 (1997).
[31] U. H. E. Hansmann, Y. Okamoto, and F. Eisenmenger, Chem. Phys. Lett. **259**, 321 (1996).
[32] D. L. Freeman, D. D. Frantz, and J. D. Doll, J. Chem. Phys. **97**, 5713 (1992).
[33] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 1987).
[34] M. Eleftheriou, D. Kim, J. D. Doll, and D. L. Freeman, Chem. Phys. Lett. (to be published).
[35] S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, J. Comput. Chem. **7**, 230 (1986).
[36] D. B. Kitchen, F. Hirata, J. D. Westbrook, R. M. Levy, D. Kofke, and M. Yarmush, J. Comput. Chem. **11**, 1169 (1990).