PHYSICAL REVIEW E **76**, 016703 (2007)

Choosing weights for simulated tempering

Sanghyun Park[1] and Vijay S. Pande[1,2]
[1]*Department of Chemistry, Stanford University, Stanford, California 94305, USA*
[2]*Department of Structural Biology, Stanford University, Stanford, California 94305, USA*
(Received 6 April 2007; published 16 July 2007)

Simulated tempering is a method to enhance simulations of complex systems by periodically raising and lowering the temperature. Despite its advantages, simulated tempering has been overshadowed by its parallel counterpart, replica exchange (also known as parallel tempering), due to the difficulty of weight determination in simulated tempering. Here we propose a simple and fast method to obtain near-optimal weights for simulated tempering, and demonstrate its effectiveness in a molecular dynamics simulation of Ala$_{10}$ polypeptide in explicit solvent. We believe simulated tempering now deserves another look.

DOI: 10.1103/PhysRevE.76.016703 PACS number(s): 05.10.−a, 87.15.Aa

I. INTRODUCTION

The success of simulations of complex systems hinges on the ability to efficiently sample their configuration spaces. Due to the ruggedness of energy landscapes, however, sampling configuration spaces is often extremely time consuming. Tempering is one of many ideas for enhancing such sampling. In tempering, the temperature of the system is changed periodically. At low temperatures, the system is easily trapped in local minima, and increasing the temperature helps the system escape. By performing a random walk in the temperature space, one can achieve broad sampling at high temperatures and thorough sampling around local minima at low temperatures. Temperature changes can be controlled according to the Metropolis algorithm [1], so that in the end one obtains canonical ensembles at all the chosen temperatures.

Two types of tempering algorithm have been developed: serial and parallel. In serial tempering (better known as simulated tempering) [2,3], a single copy of the system is simulated by constant-temperature molecular dynamics or Monte Carlo simulations, and temperature is periodically changed among a predetermined set of values. In parallel tempering (also known as replica exchange) [4,5], a set of replicas of the system is simulated in parallel at different temperatures, and periodically the temperatures of a selected pair of replicas are exchanged. Here we will use the terms simulated tempering and replica exchange, instead of serial and parallel tempering.

One important difference between simulated tempering and replica exchange is that in simulated tempering one needs to assign weights to the temperatures in order to ensure uniform sampling of the temperature space, whereas in replica exchange it is by design guaranteed to be uniform. Choosing the right weights is crucial for the success of simulated tempering, but it has so far been considered a demanding task [6]. Although adaptive weighting schemes [7–9] can reduce the burden by allowing adaptive determination of weights, convergence of weights in those schemes can be slow, especially for large systems, and therefore choosing good initial weights remains an important task. The difficulty of determining weights, it seems, has been the main reason for the relative unpopularity of simulated tempering compared to replica exchange. Nevertheless, simulated tempering has the advantage of being more robust in various computing environments (parallel tempering requires many computing nodes and frequent communication among them), which motivated recent efforts to develop serial variants of replica exchange [10].

In this paper, we introduce a simple and fast method for obtaining near-optimal weights for simulated tempering. Section II briefly outlines the algorithm of simulated tempering. In Sec. III, we propose a method for choosing weights based on average energies at each temperature. An illustration is given in Sec. IV, where the proposed method is combined with an adaptive weighting scheme in a molecular dynamics simulation of Ala$_{10}$ polypeptide in explicit solvent. Some subtle issues are discussed in Sec. V, followed by conclusions in Sec. VI.

II. SIMULATED TEMPERING

Consider a system with a Hamiltonian $H(x)$, where x denotes a microstate of the system. And suppose we have selected a set of K temperatures $T_1 < \cdots < T_K$. Define a generalized Hamiltonian

$$\mathcal{H}(x,n) = \beta_n H(x) - g_n, \tag{1}$$

where $\beta_n = 1/k_B T_n$ is the nth inverse temperature and the index n can take values $1,\ldots,K$. The generalized partition function is then given as

$$Z = \sum_n \int dx\, e^{-\mathcal{H}(x,n)} = \sum_n Z_n e^{g_n}, \tag{2}$$

where Z_n is the partition function of the system at T_n. That is, we have a generalized ensemble in which the canonical ensembles for the given temperatures are weighted differently; e^{g_n} is the weight for the nth temperature, and g_n is the corresponding logarithmic weight. Hereafter, we will simply refer to g_n as the weight.

With this generalized ensemble, simulated tempering is typically performed as follows [2,3]. A constant-temperature simulation of the system is started (using molecular dynamics or Monte Carlo simulation) at one of the chosen temperatures $T_1,\ldots,T_K$. Then, after every certain period of time, an up or down (with probability 1/2) transition is attempted. (Down transitions from T_1 and up transitions from T_K are automatically rejected.) Transitions are accepted or rejected according to the Metropolis criterion [1]; namely, when the current microstate is x, an $m \to n$ transition is accepted with probability $\min\{1, e^{-\Delta\mathcal{H}_{m\to n}(x)}\}$, where

$$\Delta\mathcal{H}_{m\to n}(x) \coloneqq \mathcal{H}(x,n) - \mathcal{H}(x,m) = (\beta_n - \beta_m)H(x) - (g_n - g_m). \tag{3}$$

Notice that it is only the relative weights (difference of weights) that matter for simulated tempering.

Without proper weighting, the random walk on the temperature space will be more or less confined within a subspace instead of exploring the entire space. Typically, in simulated tempering we want the system to explore all the given temperatures uniformly, and thus the optimal weights are the ones that lead to such uniform sampling. That is, we want to find the weights that yield

$$Z_n e^{g_n} = Z_m e^{g_m} \tag{4}$$

or

$$g_n - g_m = \beta_n F_n - \beta_m F_m, \tag{5}$$

where F_n is the free energy of the system at T_n. In the following, we show that there exists a simple and fast method to estimate such optimal weights.

III. HOW TO CHOOSE WEIGHTS

Assuming that we allow nearest-neighbor transitions only, the determination of weights boils down to the determination of the relative weights between neighboring temperatures: $g_{n+1} - g_n$ for $n = 1, \ldots, K-1$. Without loss of generality, therefore, we can focus on the problem of finding the relative weight $g_2 - g_1$ between T_1 and T_2. In order to come up with an estimate of $g_2 - g_1$, we notice the property of detailed balance: the optimal weights must yield equal acceptance ratios for the $1 \to 2$ and $2 \to 1$ transitions.[1] Finding a relative weight that exactly satisfies this property is difficult; instead, we seek to find one that satisfies the property typically.

Acceptance probabilities are determined by $\Delta\mathcal{H}$ shown in Eq. (3). For the $1 \to 2$ transition, $H(x)$ is sampled from the canonical ensemble at T_1. And, as an approximation, we choose the average energy $E_1 \coloneqq \langle H \rangle_1$ to represent the typical value sampled, where $\langle \cdot \rangle_1$ denotes a canonical ensemble average at T_1. For the $1 \to 2$ transition, therefore, we have

$$\Delta\mathcal{H}_{1\to 2} \overset{\text{typically}}{\approx} (\beta_2 - \beta_1)E_1 - (g_2 - g_1). \tag{6}$$

Similarly, for the $2 \to 1$ transition the typical value of $H(x)$ is considered to be $E_2 \coloneqq \langle H \rangle_2$, which leads to

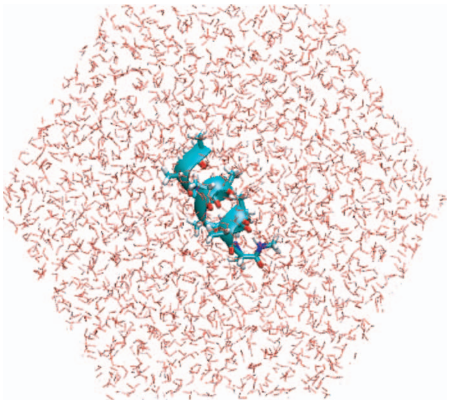

FIG. 1. (Color online) Ala$_{10}$ solvated in a rhombic dodecahedron box of water. Made with VMD [11].

$$\Delta\mathcal{H}_{2\to 1} \overset{\text{typically}}{\approx} (\beta_1 - \beta_2)E_2 - (g_1 - g_2). \tag{7}$$

By imposing detailed balance $\Delta\mathcal{H}_{1\to 2} = \Delta\mathcal{H}_{2\to 1}$ on these typical values, we obtain

$$g_2 - g_1 \approx (\beta_2 - \beta_1)\frac{E_1 + E_2}{2}, \tag{8}$$

which is generalized to

$$g_{n+1} - g_n \approx (\beta_{n+1} - \beta_n)\frac{E_n + E_{n+1}}{2} \tag{9}$$

for $n = 1, \ldots, K-1$. The weights are thus determined from the average energies $E_1, \ldots, E_K$, which are to be estimated from relatively short simulations. This is the method we propose for choosing weights for simulated tempering.

IV. ILLUSTRATION: Ala$_{10}$ IN EXPLICIT SOLVENT

To examine the effectiveness of the proposed method of weight determination, we performed a simulated tempering molecular dynamics simulation of Ala$_{10}$ polypeptide in explicit solvent. Ala$_{10}$ in an α-helical conformation, capped with acetyl and N-methyl groups, was solvated in a rhombic dodecahedron box of 2093 TIP3P [12] water molecules (Fig. 1). With position restraints applied to Ala$_{10}$, the system was equilibrated for 100 ps using the Nosé-Hoover thermostat [13,14] and the Berendsen barostat [15] at 300 K and 1 atm. During this equilibration, the volume of the system converged to 65.8 nm^3. In all the subsequent simulations, the volume was fixed at this value and no restraint was used. We used the GROMACS molecular dynamics program [16] with the AMBER 03 force field [17]. Electrostatic forces were treated with the reaction field method (cutoff distance 1 nm). All the bonds involving hydrogen atoms were constrained with LINCS [18], which allowed a 2 fs time step. The computing time was 5 h per nanosecond on a Pentium D 2.8 GHz processor.

For the ten temperatures ($T_1 = 296$ K,…,$T_{10} = 332$ K) listed in Table I, we determined the weights as follows. Starting with the final coordinates of the above simulation and velocities sampled from the Boltzmann distribution at T_1, the

---

[1] This is true because otherwise the sampling on the temperature space would not be uniform. If we allow transitions between temperatures that are not nearest neighbors, however, uniform sampling may be achieved without satisfying detailed balance.

TABLE I. Temperatures and weights. Initial weights were determined by Eq. (9). Final weights were obtained after 100 steps (100 ns) of adaptive weighting. Also listed are the frequencies at which each temperature was sampled over the entire course of the simulated tempering run.

Temperature (K)	Initial weight	Final weight	Frequency
296	−1843.8	−1843.4	0.086
300	−1399.8	−1399.5	0.093
304	−970.0	−969.5	0.098
308	−553.2	−552.7	0.098
312	−148.8	−148.5	0.101
316	243.6	243.6	0.103
320	624.5	624.1	0.102
324	994.0	993.4	0.102
328	1352.5	1352.0	0.105
332	1701.0	1700.4	0.111

TABLE II. Acceptance ratios, over the entire course of the simulated tempering run.

Transition	Acceptance ratio	Transition	Acceptance ratio
296→300	0.49	300→296	0.45
300→304	0.49	304→300	0.47
304→308	0.47	308→304	0.48
308→312	0.49	312→308	0.48
312→316	0.50	316→312	0.50
316→320	0.49	320→316	0.52
320→324	0.49	324→320	0.54
324→328	0.50	328→324	0.53
328→332	0.51	332→328	0.52

system was equilibrated for 10 ps at T_1. Then, with the final coordinates from the latter simulation and velocities sampled from the Boltzmann distribution at T_2, the system was equilibrated for 10 ps at T_2, and so forth. From these simulations (10 ps per temperature), we calculated the average potential energy at each temperature, and by using Eq. (9) determined the weights, which are listed as initial weights in Table I. Notice that Eq. (9) determines the weights g_n up to an arbitrary constant. In this illustration, we simply choose the constant such that $\Sigma_{n=1}^{K} g_n = 0$.

Starting with these weights, we performed simulated tempering for 100 ns. Temperature transitions were attempted every 1 ps, and the weights were updated every 1 ns according to the adaptive weighting scheme of Ref. [9]. A velocity rescaling scheme [19] was used so that kinetic energy terms cancel out in calculating $\Delta\mathcal{H}$ in Eq. (3); thus, potential energy alone plays the role of Hamiltonian. The potential energy distributions at each temperature are shown in Fig. 2, which shows reasonable overlaps between neighboring temperatures.

The results of the simulated tempering run are summarized in Tables I and II and Figs. 3 and 4. We note that the system was able to explore all the given temperatures even before any update of weights (Fig. 3) and that the weights did not change considerably over the course of 100 adaptive weighting steps [Table I and Fig. 4(a)]. We conclude that the initial weights estimated with Eq. (9) were already nearly optimal.

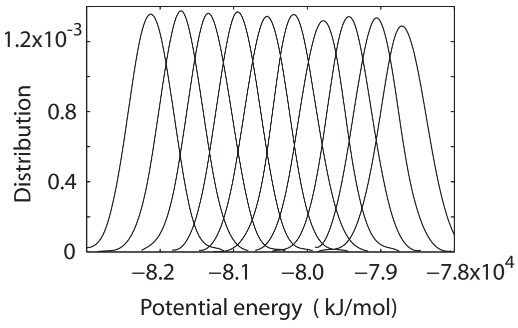

FIG. 2. Distribution of potential energy at each temperature, obtained from the entire simulated tempering run. From left to right, T_1=296 K, ..., T_{10}=332 K. Smooth curves were produced by the cubic spline interpolation of histograms of 10 bins.

This conclusion is further supported by the acceptance ratio and the histogram entropy. The acceptance ratio fluctuates around 0.5 and does not show any significant improvement as the adaptive weighting progresses [Fig. 4(b)]. Furthermore, acceptance ratio is more or less uniform over the different transitions (Table II). The histogram entropy is defined as

$$H_{\text{hist}} := -\sum_{n=1}^{K} \frac{C_n}{C_{\text{tot}}} \ln \frac{C_n}{C_{\text{tot}}}, \tag{10}$$

where C_n is the number of times that the nth temperature is sampled and $C_{\text{tot}} := \Sigma_{n=1}^{K} C_n$. It measures the degree of unifor-

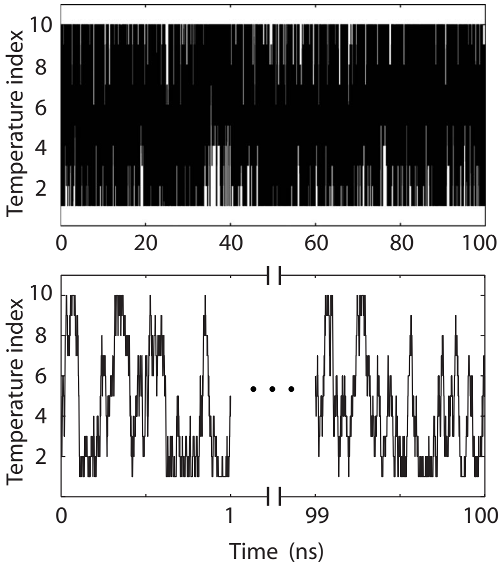

FIG. 3. Change of temperature during the simulated tempering run. Top: the entire course. Bottom: the first and the last nanosecond.

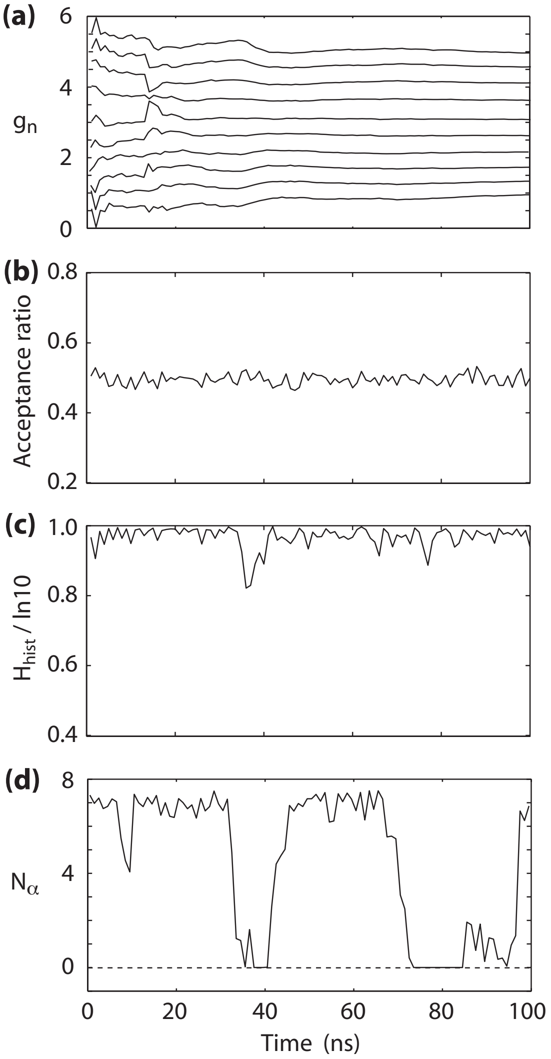

FIG. 4. Simulated tempering with adaptive weighting. (a) The weights, from bottom to top, $g_1, \ldots, g_{10}$. The ten graphs were vertically shifted relative to each other in order to make fluctuations visible. For the actual values of the weights, see Table I. (b) Acceptance ratio over all transitions for every 1 ns block. Table II lists acceptance ratios computed separately for different transitions. (c) Histogram entropy H_{hist} computed for every 1 ns block. (d) Number of α-helical residues, averaged over each 1 ns block.

mity of a histogram; it ranges from zero when only a single temperature is sampled to the maximal value of $\ln K$ when the sampling is perfectly uniform. As shown in Fig. 4(c), H_{hist} fluctuates near the maximal value throughout the entire course of simulation, showing no significant improvement during the adaptive weighting steps.

The number of α-helical residues, N_α, computed with DSSP [20] is shown in Fig. 4(d). Complete α-helical conformations of Ala$_{10}$ have $N_\alpha=8$, and complete coil conformations have $N_\alpha=0$. During the simulated tempering run of 100 ns, the Ala$_{10}$ peptide went through two rounds of transitions between helix and coil. In coil conformations, the system tends to spend more time at high temperatures, which causes slight decreases in H_{hist} [Fig. 4(c)].

V. DISCUSSION

In the above illustration, we have demonstrated that near-optimal weights can be obtained from average energies. Without this method, it would be a very demanding task to achieve convergence of weights through adaptive weighting. Ala$_{10}$ in explicit solvent was chosen as an example to highlight this point. For a smaller system, such as Ala$_{10}$ in vacuum [9], target weights may be relatively close to each other, and an adaptive weighting scheme starting with a naive guess (e.g., $g_n=0$ for $n=1,\ldots,K$) may work well. In the present example, however, target weights are separated by large amounts (Table I); without a good initial guess, any adaptive weighting scheme would require a considerable number of adaptation steps for convergence.

The number of temperatures needed for simulated tempering depends on the temperature range one chooses to include and the spacing between temperatures. But, the spacing is more or less dictated by the system itself, because energy distributions at neighboring temperatures must have significant overlap in order to produce reasonable acceptance ratios. This is also true for replica exchange; without significant overlap between energy distributions, exchange attempts are rarely accepted. As the system size grows, the widths of energy distributions increase as $N^{1/2}$, where N is the total number of the degrees of freedom. On the other hand, the separation between energy distributions increases as $N\Delta T$, where ΔT is the distance between neighboring temperatures. Therefore, in order to sustain significant overlap between energy distributions, the temperature spacing must scale as $\Delta T \sim N^{-1/2}$. That is, smaller spacing is needed for larger systems. Notice that this is required by the tempering algorithm itself, not by our method of weight determination.

Our method is based on average energies estimated from short simulations. Will it become useless if average energies take too long to converge? The answer is no. Let us consider a hypothetical situation in which, due to slow relaxation, average energies converge from E_n' to E_n'' over the course of $\sim$100 ns or even longer. Short ($\sim$10 ps) simulations will yield average energies close to E_n', not E_n''. These are, of course, not the true average energies, but they are the ones that will lead to appropriate weights for the initial stage of simulated tempering because in the initial stage energies are sampled around E_n', not E_n''. That is, if the system goes through a slow change that affects energy, no fixed set of weights will be optimal for the entire course of simulated tempering; weights need to change in response to the change in the system. The proposed method will produce near-optimal weights for the initial stage, and an adaptive weighting scheme will adjust weights as the system goes through slow changes such as relaxation or escape from a trap.

In general, we recommend combining an adaptive weighting scheme with the proposed method of weight determination as we did in the above illustration, since adaptive weighting can be useful in two regards. First, estimates given by Eq. (9) are approximations and may need refinements. Second, the system may possess a slow dynamical component such as folding and unfolding or trapping. In such cases, as discussed above, weights need to be adjusted in response to slow changes in the system.

VI. CONCLUSIONS

We have proposed a simple method for estimating weights for simulated tempering based on average energies at each temperature. As demonstrated with Ala_{10} in explicit solvent, this method yields near-optimal weights which can be further refined by adaptive weighting schemes. Tempering simulations of large systems, such as protein in explicit solvent, have been challenging because replica exchange needs many computing nodes with frequent communication among them and because simulated tempering requires determination of weights which has so far been considered a demanding task. The proposed method alleviates the difficulty of weight determination, and we believe simulated tempering now deserves another look.

ACKNOWLEDGMENTS

We would like to thank Daniel Ensign for implementing simulated tempering in GROMACS, and John Chodera, Xuhui Huang, and Vincent Voelz for exciting discussions and comments on the manuscript. This work was supported by a grant from NSF for Cyberinfrastructure.

---

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[2] E. Marinari and G. Parisi, Europhys. Lett. **19**, 451 (1992).
[3] A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, J. Chem. Phys. **96**, 1776 (1992).
[4] C. J. Geyer and E. A. Thompson, J. Am. Stat. Assoc. **90**, 909 (1995).
[5] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[6] A. Mitsutake and Y. Okamoto, J. Chem. Phys. **121**, 2491 (2004).
[7] C. Bartels and M. Karplus, J. Comput. Chem. **18**, 1450 (1997).
[8] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).
[9] S. Park, D. L. Ensign, and V. S. Pande, Phys. Rev. E **74**, 066703 (2006).
[10] M. Hagen, B. Kim, P. Liu, R. A. Friesner, and B. J. Berne, J. Phys. Chem. B **111**, 1416 (2007).
[11] W. Humphrey, A. Dalke, and K. Schulten, J. Mol. Graphics **14**, 33 (1996).
[12] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**, 926 (1983).
[13] S. Nosé, J. Chem. Phys. **81**, 511 (1984).
[14] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).
[15] H. J. C. Berendsen, J. P. M. Postma, A. DiNola, and J. R. Haak, J. Chem. Phys. **81**, 3684 (1984).
[16] E. Lindahl, B. Hess, and D. van der Spoel, J. Mol. Model. **7**, 306 (2001).
[17] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, and T. Lee, J. Comput. Chem. **24**, 1999 (2003).
[18] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije, J. Comput. Chem. **18**, 1463 (1997).
[19] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).
[20] W. Kabsch and C. Sander, Biopolymers **22**, 2577 (1983).