Evolutionary Monte Carlo for protein folding simulations

Faming Liang[a)]
Department of Statistics and Applied Probability, The National University of Singapore, 3 Science Drive 2, Singapore 117543

Wing Hung Wong[b)]
Department of Biostatistics, HSPH, 655 Huntington Avenue, Boston, Massachusetts 02115 and Department of Statistics, Harvard University, Cambridge, Massachusetts 02138

(Received 26 February 2001; accepted 1 June 2001)

We demonstrate that evolutionary Monte Carlo (EMC) can be applied successfully to simulations of protein folding on simple lattice models, and to finding the ground state of a protein. In all cases, EMC is faster than the genetic algorithm and the conventional Metropolis Monte Carlo, and in several cases it finds new lower energy states. We also propose one method for the use of secondary structures in protein folding. The numerical results show that it is drastically superior to other methods in finding the ground state of a protein.
[DOI: 10.1063/1.1387478]

I. INTRODUCTION

In recent years the prediction of the native structure of a protein from its sequence have attracted a great deal of attention. Given a polypeptide chain and the correct molecular potential, how can one find the thermodynamically stable state of the protein? This problem has bee recognized to be "NP-complete,"[1–3] which means that this problem is not solvable in polynomial time, even for an optimal algorithm. The difficulty of the problem is that the energy landscape of the system is characterized by a multitude of local minima separated by high energy barriers. At low temperatures, traditional Monte Carlo and molecular dynamics simulations tend to get trapped in local minima. Hence, only a small fraction of the phase space is sampled, the native structure cannot be located, and the thermodynamic quantities cannot be estimated accurately.

Attempts to alleviate this difficulty have been in two directions. One direction is to search for the lowest potential energy conformation (which is believed to correspond to the native state of a protein) with powerful optimization techniques such as Monte Carlo with minimization,[4] simulated annealing,[5] and genetic algorithms.[6,7] The effectiveness of these methods have been tested with many proteins and lattice models.[8,9] One drawback is that these optimizers ignore the entropic contributions of the conformations and the thermodynamic quantities of interest cannot be estimated. The other direction is to sample the phase space with more efficient samplers such as multicanonical,[10] entropic sampling,[11] parallel tempering,[12,13] simulated tempering,[14] $1/k$-ensemble sampling,[15] chain growth algorithms,[16–19] and Metropolis algorithms with long range moves.[20,21] For a recent review, see Ref. 22.

In this paper we test our a new Monte Carlo algorithm, the evolutionary Monte Carlo (EMC) (Ref. 23) algorithm, on 2D hydrophobic–hydrophilic (HP) models.[24] This algorithm is motivated by the genetic algorithm in optimization. It works by simulating a population of Markov chains, where a different temperature is attached to each chain. The population is updated by mutation, crossover and exchange operators that preserve the Boltzmann distribution of the population. EMC possesses two attractive features as a simulation algorithm. One is that it has incorporated the extensively search ability of the genetic algorithm by evolving with crossover operators. The other one is that it has incorporated the fast mixing ability of simulated tempering by simulating along a temperature ladder. The numerical results show that EMC is a promising algorithm for simulation and optimization.

II. THE 2D HP MODEL

In the 2D HP model a protein is composed of "amino acids" of only two types: hydrophobic (H for nonpolar) and hydrophilic (P for polar). The sequence is "folded" on a two-dimensional square lattice. At each point the chain can turn 90° left, right, or continue ahead. Only the self-avoiding conformations are valid with energies $\epsilon_{HH}=-1$ and $\epsilon_{HP}=\epsilon_{PP}=0$ for interactions between noncovalently bound neighbors. The interest in the model comes from the fact that although it is very simple, it does exhibit many of the features of real protein folding.[25,26] For the 2D HP model, low energy conformations are compact with a hydrophobic core, since the H–H interactions are rewarded. The hydrophobic residues have to be buried inside to yield a low energy structure, while the hydrophilic residues are forced to the surface. This model has been used by chemists to evaluate new hypothesis of protein structure formation.[27] Also, the simplicity of the model permits a rigorous analysis of the efficiency for a folding algorithm. In fact, this model has become standards in testing efficiency of folding algorithms.

[a)] Author to whom correspondence should be addressed; Electronic mail: stalfm@nus.edu.sg

[b)] Electronic mail: wwong@hsph.harvard.edu

III. EVOLUTIONARY MONTE CARLO ALGORITHM

The algorithm we applied here is a special implementation of EMC for the 2D HP model. To apply EMC to the 2D HP model, each conformation (individual or chromosome in terms of genetic algorithms) of a protein is represented by a vector $z=(z^{(1)},\ldots,z^{(d)})$ with $z^{(i)(}\in\{0,1,2\}$, where each digit represents a torsion angle: 0, right; 1, continue and 2, left. The energy function of the conformation is denoted by $H(z)$ and the Boltzmann distribution is defined by

$$f(z)\propto\exp\{-H(z)/\tau\},$$

where τ is called the temperature of the system.

In EMC, to simulate from the target distribution $f(z)$, a sequence of distributions $f_1,\ldots,f_N$ are first constructed with

$$f_i(z)\propto\exp\{-H(z)/t_i\},\quad i=1,...,N.$$

The temperature sequence $\mathbf{t}=(t_1,\ldots,t_N)$ forms a ladder with $t_1>\ldots>t_N\equiv\tau$. Hence, $f_N(\cdot)$ turns out to be the target distribution to be sampled from. Let z_i denote a sample from $f_i(x)$, the samples $z_1,\ldots,z_N$ form a population denoted by $\mathbf{z}=\{z_1,\ldots,z_N\}$, where N is called the population size. In EMC, the Markov chain state is augmented as a population of samples instead of a single sample, and the Boltzmann distribution of the population is defined as

$$f(\mathbf{z})\propto\exp\left\{-\sum_{i=1}^{N}H(z_i)/t_i\right\}. \tag{1}$$

The population is updated by three operators: mutation, crossover, and exchange (described below).

In the mutation operator, an individual, say z_m, is first equally likely selected from the current population $\mathbf{z}$. Then z_m is mutated to a new chromosome z'_m. A new population is formed by replacing z_m by z'_m, that is, $\mathbf{z}'=\{z_1,\ldots,z_{m-1},z'_m,z_{m+1},\ldots,z_N\}$. The new population is accepted with probability $\min(1,r_m)$ according to the Metropolis–Hastings rule,[28,29]

$$r_m=\frac{f(\mathbf{z}')}{f(\mathbf{z})}\,\frac{T((\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}$$

$$=\exp\{-(H(z'_m)-H(z_m))/t_m\}\frac{T(\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}, \tag{2}$$

where $T(\cdot|\cdot)$ denotes the transition probability between two populations. If the proposal is accepted, the current population $\mathbf{z}$ is replaced by $\mathbf{z}'$, otherwise, $\mathbf{z}$ is unchanged.

The mutation operators used in this paper include a k-point mutation, a three-bead flip, a crankshaft move, and a rigid rotation. In k-point mutations, k bits are randomly chosen from the individual z_m, and their values are replaced by the ones sampled uniformly from the set {0,1,2}. The k is also a random variable, of which the value ranges from 1 to d. When $k=d$, the operator produces a completely new random conformation independent of the current one, and it effectively prevents the population from becoming homogeneous. The k-point mutation operator is symmetric in the sense that $T(\mathbf{z}|\mathbf{z}')=T(\mathbf{z}|\mathbf{z}')$, and r_m in Eq. (2) is reduced to the conventional Metropolis ratio. The other mutation operators are identical to the local moves used in Ref. 30, and they are illustrated in Fig. 1. For example, in the crankshaft operator [Fig. 1(b)], the two crankshaft structures can be represented as 2002 and 0220 in our coding scheme, respectively. We first search for all of the crankshaft structures from z_m and denote the total number of them by c_m, and then randomly choose one crankshaft uniformly on 1 to c_m to mutate by reversing (i.e., change 0220 and 2002 and vice versa). The transition probability ratio is then $T(\mathbf{z}|\mathbf{z}')/\mathbf{T}(\mathbf{z}'|\mathbf{z})=c'_m/c_m$, where c'_m denotes the total number of local crankshaft structures in z'_m. The other operators are performed in a similar manner.

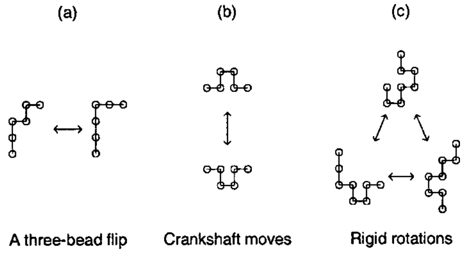

FIG. 1. Mutation operators used in 2D HP models. The circle denotes a residue, hydrophobic or hydrophilic. (a) A three-bead flip. (b) Crankshaft moves. (c) Rigid rotations.

In the crossover operator, different offspring are produced by a recombination of parental chromosomes randomly selected from the population. For example, z_a and z_b are selected as the parental chromosomes. Without loss of generality, we assume that $H(z_a)>H(z_b)$. Two "offspring" z'_a and z'_b are generated as follows. First an integer crossover point c is drawn uniformly on $\{1,...,d\}$, then z'_a and z'_b are constructed by swapping the genes of the two parental chromosomes to the right of the crossover point. The following diagram illustrates a 1-point crossover operator,

$$\begin{array}{ll}(z_a^{(1)},\ldots,z_a^{(d)}) & (z_a^{(1)},\ldots,z_a^{(c)},z_b^{(c+1)},\ldots,z_b^{(d)}) \\ & \Rightarrow \\ (z_b^{(1)},\ldots,z_b^{(d)}) & (z_b^{(1)},\ldots,z_b^{(c)},z_a^{(c+1)},\ldots,z_a^{(d)})\end{array}$$

where c is called a crossover point. If there are $k(k>1)$ crossover points, it is called a k-point crossover. In this paper, we use $k=1$ or 2.

A new population is formed by replacing the selected parental chromosomes with the new "offspring," z_a is replaced by the offspring with the higher energy and z_b is replaced by the other one. The new population is accepted with probability $\min(1,r_c)$ according to the Metropolis–Hastings rule,

$$r_c=\frac{f(\mathbf{z}')}{f(\mathbf{z})}\,\frac{T(\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}$$

$$=\exp\{-(H(z'_a)-H(z_a))/t_a-(H(z'_b)$$

$$-H(z_b))/t_b\}\frac{T(\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}, \tag{3}$$

where $T(\mathbf{z}'|\mathbf{z})=P((z_a,z_b)|\mathbf{z})P((z'_a z'_b)|(z_a,z_b))$, $P((z_a,z_b)|\mathbf{z})$ denotes the selection probability of (z_a,z_b)

from the population $\mathbf{z}$, and $P((z'_a,z'_b)|(z_a,z_b))$ denotes the generating probability of (z'_a,z'_b) from the parental chromosomes (z_a,z_b).

Throughout this paper, the parental chromosomes are chosen as follows. The fist chromosome $z_a(z_b)$ is selected according to a roulette wheel procedure with Boltzmann weights, that is, $z_a(z_b)$ is selected with a probability,

$$w(z)=\frac{1}{C(\mathbf{z})}\exp\{-H(z)/t_s\},$$

where $C(\mathbf{z})=\Sigma_{i=1}^{N}\exp\{-H(z_i)/t_s\}$, and t_s is called a selection temperature, which takes a value around t_N. It may be the same with t_N but not necessary. The second chromosome $z_b(z_a)$ is selected randomly from the rest of the population. The selection probability of (z_a,z_b) from $\mathbf{z}$ is then

$$P((z_a,z_b)|\mathbf{z})=\frac{1}{(N-1)C(\mathbf{z})}\times[\exp\{-H(z_a)/t_s\}+\exp\{-H(z_b)/t_s\}].$$

$P(z'_a,z'_b|\mathbf{z}')$ can be calculated similarly. Note that the 1-point and 2-point crossover operators are both symmetric, i.e., $P((z_a,z_b)|(z'_a,z'_b))=P((z'_a,z'_b)|(z_a,z_b))$.

The exchange operator is the same with that of parallel tempering. Given the current population $\mathbf{z}$ and the attached temperature ladder $\mathbf{t}$, we try to make an exchange between z_i and z_j without changing the t's, i.e., initially we have $(\mathbf{z},\mathbf{t})=(z_1,t_1,\ldots,z_i,t_i,\ldots,z_j,t_j,\ldots,z_N,t_N)$ and we want to change it to $(\mathbf{z}',\mathbf{t})=(z_1,t_1,\ldots,z_j,t_i,\ldots,z_i,t_j,\ldots,z_N,t_N)$. The new population is accepted with probability $\min(1,r_e)$ according to the Metropolis–Hastings Rule,

$$r_e=\frac{f(\mathbf{z}')}{f(\mathbf{z})}\frac{T(\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}=\exp\left\{(H(z_i)-H(z_j))\left(\frac{1}{t_i}-\frac{1}{t_j}\right)\right\}\frac{T(\mathbf{z}|\mathbf{z}')}{T(\mathbf{z}'|\mathbf{z})}. \qquad (4)$$

The exchange operator usually only performs on the individuals with neighboring temperatures, i.e., $|i-j|=1$.

With the operators described above, one iteration of EMC consists of the following two steps:

(1) Apply either mutation or crossover operator to the population with probability p_m and $1-p_m$, respectively, where p_m is called a mutation rate.

(2) Try to exchange z_i with z_j for $N-1$ pairs (i,j) with i being sampled uniformly on $\{1,\ldots,N\}$ and $j=i\pm1$ with probability $q_{i,j}$, where $q_{i,i+1}=q_{i-1,i}=0.5$ and $q_{1,2}=q_{N,N-1}=1$.

In the mutation step, each chromosome of the population is mutated independently. In the crossover step, about $N/2$ pairs of chromosomes are chosen to mate. The crossover operator works in an iterative way, that is, each time the operator works on a new population which was just updated by the preceding operator. In our experience, the mutation operator provides a good local exploration around a local minimum, while the crossover operator provides a much global exploration over the whole conformation space. The algorithm has three user-set parameters, namely N, $\mathbf{t}$, and p_m. Issues about choice of them can be found in Ref. 23. Roughly speaking, we should choose the population size comparable (at least) to the length of the chromosome; choose the highest temperature such that the (anticipated) energy barriers can be overcome easily by a Metropolis–Hastings move, choose the temperature ladder such that the acceptance probability of the exchange operations is moderate, and choose the mutation rate to balance the local and global searches, e.g., a value between 0.2 and 0.4.

TABLE I. Test of ergodicity of EMC simulations. The target is the Boltzmann distribution at the lowest temperature level $t_N=0.5$.

	Energy						
	0	−1	−2	−3	−4	−5	−6
Exact[a]	0.0170	0.0602	0.1295	0.2180	0.2643	0.1983	0.1127
EMC[b]	0.0171	0.0618	0.1286	0.2190	0.2653	0.1981	0.1101
SD[c]	0.0003	0.0016	0.0016	0.0023	0.0016	0.0022	0.0033

[a] Mass of each valid energy value from the exact equilibrium distribution.
[b] Percentage of time spent in each energy value by the EMC simulations.
[c] SD: standard deviation of the EMC estimator based on 10 runs.

The structure of EMC is very flexible. Setting $p_m=1$, the algorithm is reduced to parallel tempering.[12,13] Setting $p_m=1$ and $N=1$, the algorithm is reduced to the conventional single-chain Metropolis–Hastings algorithm.

IV. A TESTING EXAMPLE

The sequence used for this example is: HHPHPHPHPPHPH, which is identical to sequence 1 of Ref. 31. It has a unique native conformation with energy −6. EMC was run 10 times independently. Each run consists of 50 000 iterations. The parameters are set as follows: the population size $N=50$; the lowest temperature is 0.5, the highest temperature is 10, and the intermediate temperatures are equally spaced between 0.5 and 10; the selection temperature is 0.5; and the mutation rate is 0.25. In the mutation step, all individuals in the current population are independently subject to an attempt of mutation, and the proportions of the k-point mutation, three-bead flip, crankshaft move and rigid rotation are 1/2, 1/6, 1/6 and 1/6, respectively. In the crossover step, $N/2$ chromosome pairs are selected to mate. This parameter setting may not be optimal, but it works well for this example. In the following examples, only the population size, temperature ladder, and selection temperature vary with lengths of sequences, and the other parameters are kept unchanged.

Table I shows the Boltzmann distributions ($t_N=0.5$) generated by EMC and the exact equilibrium distribution computed by an exhaustive enumeration for this testing example. The comparison indicates that EMC is ergodic, and the 10 runs have achieved agreement between the Monte Carlo estimate and the exact equilibrium distribution to within a few percent.

V. A COMPARISON BETWEEN THE METHODS

In this section, EMC is compared to two other commonly used protein folding algorithms, namely, the genetic algorithm and Metropolis Monte Carlo. The latter two algorithms have been implemented in Ref. 9. In protein folding

TABLE II. Comparison of EMG with the genetic algorithm (GA) and Metropolis Monte Carlo (MC). For each sequence, the three algorithms were all run 5 times independently, the lowest energy values achieved during the most efficient run were reported in the respective columns together with the number of valid conformations scanned before that value was found.

Length[a]	Putative ground energy	EMC[b]	GA[c]	MC[d]
20	−9	−9 (9,374)	−98 (30,492)	−9 (292,443)
24	−9	−9 (6,929)	−9 (30,491)	−9 (2,492,221)
25	−8	−8 (7,202)	−8 (20,400)	−8 (2,694,572)
36	−14	−14 (12,447)	−14 (301,339)	−13 (6,557,189)
48	−23	−23 (165,791)	−22 (126,547)	−20 (9,201,755)
50	−21	−21 (74,613)	−21 (592,887)	−21 (15,151,203)
60	−36	−35 (203,729)	−34 (208,781)	−33 (8,262,338)
64	−42	−39 (564,809)	−37 (187,393)	−35 (7,848,952)

[a] The length denotes the number of residues of the sequence.

[b] For sequence 20, 24, and 25, EMC was run for 5000 iterations with the population size 100, the highest temperature 20, the lowest temperature 0.3, and the selection temperature 0.3. For the other sequences, EMC was run for 1000 iterations with the population size 500, the highest temperature 20, the lowest temperature 0.3, and the selection temperature 0.5.

[c] The results of the genetic algorithm reported in Ref. 9. The GA was run with the population size 200 for 300 generations.

[d] The results of Metropolis Monte Carlo reported in Ref. 9. Each run of MC consists of 50 000 000 steps.

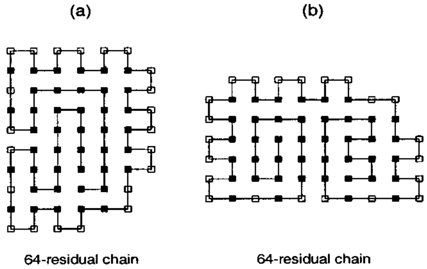

FIG. 2. Two local minimum energy conformations of energy −39 found by EMC for the 64-mer sequence (without the use of secondary structures). (a) Conformation 1; (b) Conformation 2. Solid squares represent hydrophobic residues whereas open squares represent hydrophilic residues.

simulations, the dominant factor is the energy evaluation, which is performed once for each valid conformation. EMC is not significantly more costly per step than the genetic algorithm, since most of the mutation and crossover operators are designed to be symmetric, and the corresponding transition probabilities need not to be evaluated. In the exchange step, no energy evaluation is performed. Hence, the main factors to be compared are the number of energy evaluations needed to find one of the lowest energy conformations, and the lowest energy attained for a given number of energy evaluations.

The three algorithms were compared for the sequences given in Ref. 9. The results were summarized in Table II. For sequences of length 20, 24, 25, 36, and 50, EMC was faster than the genetic algorithm and Metropolis Monte Carlo in finding the putative ground energy states. The computational amounts used by EMC were only about 10%–30% of that used by the genetic algorithm, and 1%–3% of that used by Metropolis Monte Carlo. For sequences of length 48, 60, and 64, EMC also found the same energy states with smaller computational amounts than the generic algorithm and Metropolis Monte Carlo. With slightly larger computational amounts, EMC found some new lower energy states as reported in Table II. Two local minimum energy conformations of energy −39 found by EMC for the 64-mer sequence were shown in Fig. 2. Both are in globular shapes with compact hydrophobic cores being buried by hydrophilic residues. In summary, Table II shows that EMC has made a significant improvement over the genetic algorithm and Metropolis Monte Carlo in finding low energy states for a protein. We note that for the 60-mer sequence a putative ground state was found by the pruned-enriched Rosenbluth method (PERM).[19] A direct comparison of PERM with EMC, Metropolis Monte Carlo and the genetic algorithm is unfair, since the latter three algorithms only perform ‘‘blind’’ searches over the whole conformation space. In contrast, PERM makes use of more information of the sequence when it folds a protein. PERM may build up its chain from any part of a sequence, e.g., a subsequence of hydrophobic residues. PERM (Ref. 19) argues for this idea that real proteins have folding nuclei,[32] and it should be most efficient to start from such a nucleus. The issue about the use of the subsequence information in protein folding simulations will be further discussed in the next section.

VI. The USE OF SECONDARY STRUCTURES IN PROTEIN FOLDING

Levinthal[33] and Wetlaufer[34] pointed out that proteins fold much too fast (by at least tens of orders of magnitude) to involve an exhaustive search. This is the so-called Levinthal paradox: how can a protein find a native state without a globally exhaustive search? Experiments show that there exists ‘‘cooperativity’’ in protein folding, i.e., a protein folds to its native state according to a relatively small number of ‘‘pathways,’’ in other words, it folds by a specific sequence of molecular events.[35.36] The cooperativity are mainly driven by two types of interactions:[31,37] (i) the local interaction by which each individual tetrapeptide in the sequence finds a hydrogen-bonded helical conformation, and (ii) the nonlocal interaction by which a compact hydrophobic (H) core is formed.

Motivated by the above observations, we use the following steps to speed up the simulations of protein folding:

(1) Identify the subsequences which will possibly fold to secondary structures[38] in the native state of a given protein.
(2) Perform sampling on the constrained conformation space where some subsequences are subject to possible secondary structures.

However, the simulation performed on the constrained conformation space may lead to a biased estimate if we are interested in the thermodynamic properties of protein folding. An ergodic simulation with the use of secondary structures is described as follow.

For the 2D HP model, the possible secondary structures folded by a subsequence of hydrophobic residues are illustrated in Figs. 3(a), 3(b), and 3(c), which correspond to beta,

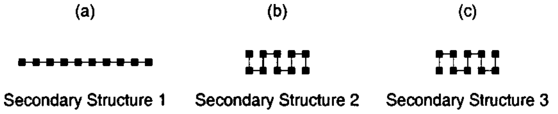

FIG. 3. Secondary structures folded by a subsequence of hydrophobic residues. (a) Extended sheet. (b) Helix with direction 1. (c) Helix with direction 2.

alpha (with direction 1), and alpha (with direction 2) structures of a real protein, respectively. However, the total number of self-avoiding structures that could be folded by the subsequence may be huge. For example, it is 4067 for a subsequence of 10 residues. An essentially arbitrary distribution can be assigned to these self-avoiding structures with each structure having a nonzero mass value. For example, we assign that each structure shown in Fig. 3 has a mass value $1/3-\epsilon$, and all other structures have an equal mass $3\epsilon/(4067-3)$, where ϵ is a small value chosen by users. The mass function will then work as a proposal transition function for the move of the block of residues, and the resulting simulation will be ergodic. In fact, the block move can be incorporated easily into EMC as a mutation operator. Let ϵ tend to 0, the simulation reduces to sampling on the constrained conformation space.

By sampling on the constrained conformation space, we fold the 48-mer and 64-mer sequences and a new 85-mer sequence[39] rapidly to their putative ground energy states. The primary sequences of them are given as follows:

(48) PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH;

(64) HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHPPHHPPHHPPHPHPHHHHHHHHHHHH;

(85) HHHHPPPPHHHHHHHHHHHHPPPPPPHHHHHHHHHHHHPPPHHHHHHHHHHHHPPPHHHHHHH
HHHHHPPPHPPHHPPHHPPHPH.

The computational results were summarized in Table III. The putative ground states found by the constrained EMC sampler for the 48-mer and 64-mer sequences are shown in Fig. 4. Using other algorithms, including Metropolis Monte Carlo,[9] the genetic algorithm,[9,39] and PERM,[18,19] the putative ground energy states of the 64-mer sequence were never found. The authors[19] commented that this sequence acts as a bottleneck for PERM due to the lack of a folding center in the protein. The 85-mer sequence has a putative ground energy of −52. The genetic algorithm[39] failed to fold the sequence to one of its putative ground energy states, and the lowest energy attained was −47, which is far from the putative ground energy value. PERM was not tried on this sequence. For this sequence, EMC was tried with different constraints. For example,

(a) 11–18, 29–36, 44–51, 59–66;

(b) 9–18, 27–36, 42–51, 57–66;

(c) 9–20, 27–38, 42–53, 57–68.

With constraint (a), the EMC folds the sequence rapidly to many states with energy −51 [e.g., Figs. 5(a) and 5(b)]. With slightly stronger constraints (b) and (c), EMS folds the sequence rapidly to putative ground states. For example, in one run with constraint (b), only 44029 valid conformations are scanned before one putative ground state was found; and in one run with constraint (c), only 17794 valid conformations are scanned before one putative ground state was found. Two representatively putative ground conformations found by EMS are shown in Fig. 6, and the other putative ground conformations found by EMC are only their rotated versions. Note that the conformation shown in Fig. 6(a) is identical to

TABLE III. Protein folding simulations with the use of secondary structures.

Length[a]	Putative ground energy	EMC[b]
48	−23	−23 (53,263)
64	−42	−42 (77,287)
85	−52	−52 (44,029)

[a] In the sequence (48), the subsequence (residues 17–26) is constrained to the secondary structure. In the sequence (64), the subsequences (residues 1–0, 55–64) are constrained to the secondary structures. In the sequence (85), the subsequences (residues 9–18, 27–36, 42–51, 57–66) are constrained to the secondary structures.

[b] EMC was run 5 times independently with the population size 500, the highest temperature 20, the lowest temperature 0.5, and the selection temperature 0.5. The reported values are the lowest energy achieved during the most efficient run and the number of valid conformations scanned before that value was found.

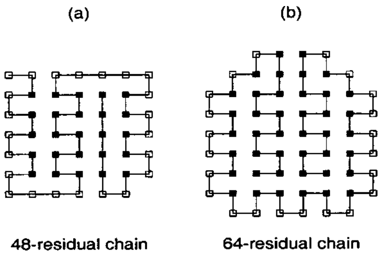

FIG. 4. The putative ground conformations found by EMC with secondary structure constraints. (a) The putative ground conformation of energy −23 of the 48-mer sequence with the subsequence (residues 17–26) being constrained to the secondary structures. (b) The putative ground conformation of energy −42 of the 64-mer sequence with the subsequences (residues 1–10 and 55–64) being constrained to the secondary structures.

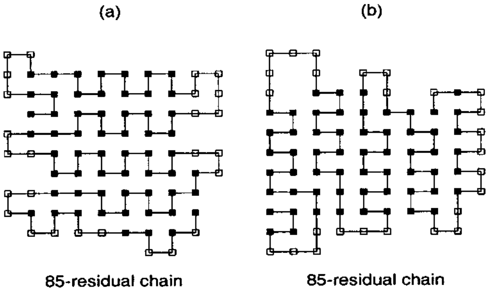

FIG. 5. Two local minimum conformations of energy −51 found by EMC for the 85-mer sequence with constraint (a) where the subsequences (residues 11–18, 29–36, 44–51, 59–66) are constrained to the secondary structures.

that originally designed in Ref. 39. From the three examples, we conclude that the use of the secondary structure constraints substantially eases the search for the lower energy states in simulations, but it also limits the number of ground conformations that could be found by EMC. For example, the putative ground conformations [Figs. 6(a) and 6(b)] found by EMC for the 85-mer sequence have the identical hydrophobic cores.

For a thorough comparison, we also applied the constrained EMC sampler to the 60-mer sequence. It folds the sequence rapidly to some states of energy −35. However, if we restrict the subsequence (residues 33–44) to a local structure as shown in Fig. 7(a), EMC folds rapidly to one putative ground state of energy −36 [Fig. 7(b)]. In one run, only 40334 valid conformations were scanned before the putative ground state was found. This experiment suggests that an appropriately large number of secondary structures should be used for an efficient simulation to accommodate various local structures of proteins. These structures may be determined by a survey for the frequencies of secondary structures being adopted by real proteins.

The necessity of the use of local structures in protein folding simulations can be justified by comparing Fig. 2 and Fig. 4(b). Although EMC has folded the 64-mer sequence to globular shapes with and without the use of secondary structures, the transition from a local minimum energy state, e.g., the conformation shown in Fig. 2(a) or 2(b), to the putative ground state may cost an extremely long time if secondary structures were not used. In real protein folding simulations, a procedure of secondary structure prediction may be performed first. The predicted secondary structures and their respective folding probabilities can be incorporated into the simulation, and this will substantially accelerate the simulation process of protein folding.

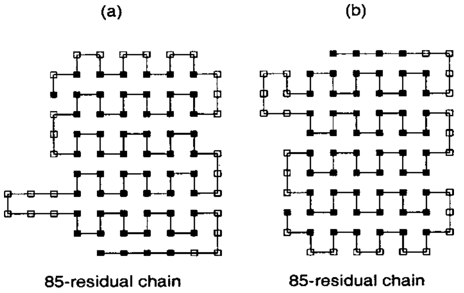

FIG. 6. Two putative ground conformations of energy −52 found by EMC for the 85-mer sequence with constraint (b) or (c). In constraint (b), the sequences (residues 9–18, 27–36, 42–51, 57–66) are constrained to the secondary structures. In constraint (c), the subsequences (residues 9–20, 27–38, 42–53, 57–68) are constrained to the secondary structures.

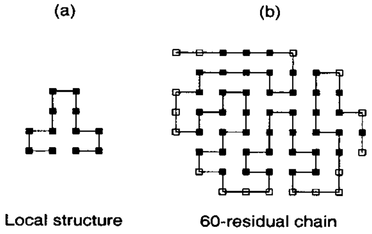

FIG. 7. A local structure (a) and a putative ground conformation (b) of energy −36 of the 60-mer sequence. The putative ground conformation was found by EMC with the subsequence (residues 33–44) being constrained to the local structure shown in (a).

VII. CONCLUSION AND DISCUSSION

We showed that the evolutionary Monte Carlo algorithm can be effectively applied to simulations of protein folding on lattice models. In all cases it did better than the genetic algorithm and Metropolis Monte Carlo, and in several cases it found new lower energy states. We also proposed one method for the use of secondary structures in protein folding. The numerical results showed that it is very successful in finding low energy states. Although we have considered only 2D HP models in this article, we should stress that the extension to 3D HP and real protein models is straightforward.

For EMC, two points needs to stress are its flexible structure and learning capability. The structure of EMC is so flexible that any efficient move developed for protein folding can be incorporated as a mutation operator or a crossover operator. For example, the long range moves[20,21] can be incorporated into EMC as mutation operators, and the resulting simulation will be more efficient for dense chains. Similarly, the systematic crossover operator[39] can also be incorporated into EMC as a crossover operator by an importance sampling procedure, by which one pair of offspring are selected from a temporary stock, where offspring produced from the same parents with various crossover points are stored. The detailed balance condition is ensured by accounting for the selection probabilities of the pair of offspring.

In general, a ‘‘learning’’ capability refers to that one is able to modify its behavioral tendency by experience. The use of population makes it possible for EMC to learn from its historical samples. In EMC, the simulation at high temperatures can help the system make a much global exploration over the whole sample space. The exchange operation (swapping of temperatures) works as a selection mechanism. A high energy conformation will, through exchange operations, be forced to climb up the temperature ladder. At high temperatures, random mutations are easily accepted and thus the high energy conformation will be eliminated from the population. While a low energy conformation will be forced to climb down the temperature ladder. At low temperatures, random mutations are difficult to accept and the low energy conformation will be stored there for a relatively long time period. According to the roulette wheel selection procedure employed by EMC, the low energy conformation will have a large probability to be selected as a parental chromosome to mate with other chromosomes to produce offspring, which will be in a large probability to resemble the parental conformations. In this sense, we say that EMC has learned from its historical samples.

A further work of interest is how to design a more efficient crossover operator for a long sequence. The crossover operators used in this paper are very effective for a sequence of short and moderate length, but are less effective for a long sequence due to a low acceptance probability, just as noticed by the authors[20,21] for long range moves. The crossover operators do improve the simulations of protein folding.

ACKNOWLEDGMENTS

F.L. was partially supported by an academic research Grant No. R-155-000-010-112. W. H. was partially supported by NSF Grant Nos. DBI-9904701/0196176 and DMS-0090166.

[1] R. Unger and J. Moult, Bull. Math. Biol. **55**, 1183 (1993).
[2] B. Berger and T. Leighton, J. Comput. Biol. **5**, 27 (1998).
[3] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, J. Comput. Biol. **5**, 423 (1998).
[4] Z. Li and H. A. Scheraga, Proc. Natl. Acad. Sci. U.S.A. **84**, 6611 (1987).
[5] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Science **220**, 671 (1983).
[6] J. H. Holland, *Adaptation in Natural and Artificial Systems* (The University of Michigan Press, Ann Arbor, 1975).
[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison–Wesley, Reading, 1989).
[8] S. R. Wilson and W. Cui, in *The Protein Folding Problem and Tertiary Structure Prediction*, edited by K. M. Merz, Jr. and S. M. Legrand (Birkhauser, Berlin, 1994), p. 43.
[9] R. Unger and J. Moult, J. Mol. Biol. **231**, 75 (1993).
[10] B. A. Berg and T. Neuhaus, Phys. Lett. B **267**, 249 (1991); Phys. Rev. Lett. **6**, 9 (1992).
[11] J. Lee, Phys. Rev. Lett. **71**, 211 (1993).
[12] C. J. Geyer, in *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, edited by E. M. Keramigas (Interface Foundations, Fairfax, 1991), p. 156.
[13] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[14] E. Marinari and G. Parisi, Europhys. Lett. **19**, 451 (1992).
[15] B. Hesselbo and R. B. Stinchcomb Phys. Rev. Lett. **74**, 2151 (1995).
[16] M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).
[17] P. Grassberger, Phys. Rev. E **56**, 3682 (1997).
[18] H. Frauenkron, U. Bastolla, E. Gerstner, P. Grassberger, and W. Nadler, Phys. Rev. Lett. **80**, 3149 (1998).
[19] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, Proteins: Struct., Funct., Genet. **32**, 52 (1998).
[20] R. Ramakrishnan, B. Ramachandran, and J. F. Pekny, J. Chem. Phys. **106**, 2418 (1997).
[21] J. M. Deutsch, J. Chem. Phys. **106**, 8849 (1997).
[22] U. H. E. Hansmann and Y. Okamoto, Curr. Opin. Struct. Biol. **9**, 177 (1999).
[23] F. Liang and W. H. Wong, Statistica Sinica **10**, 317 (2000).
[24] K. A. Dill, Biochemistry **24**, 1501 (1985).
[25] K. F. Lau and K. A. Dill, Proc. Natl. Acad. Sci. U.S.A. **87**, 638 (1990).
[26] G. M. Crippen, Biochemistry **30**, 4232 (1991).
[27] A. Sali, E. Shahknovich, and M. Karplus, Nature (London) **369**, 248 (1994).
[28] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[29] W. K. Hastings, Biometrika **57**, 97 (1970).
[30] H. S. Chan and K. A. Dill, J. Chem. Phys. **99**, 2116 (1993); **100**, 9238 (1994).
[31] R. Miller, C. A. Danko, M. J. Fasolka, and A. C. Balazs, J. Chem. Phys. **96**, 768 (1992).
[32] R. R. Matheson and H. A. Scheraga, Macromolecules **11**, 819 (1978).
[33] C. Levinthal, J. Chem. Phys. Phys. Biol. **65**, 44 (1968).
[34] D. B. Wetlaufer, Proc. Natl. Acad. Sci. U.S.A. **70**, 697 (1973).
[35] T. E. Creighton, Prog. Biophys. Mol. Biol. **33**, 231 (1978).
[36] P. S. Kim and R. L. Baldwin, Annu. Rev. Biochem. **59**, 631 (1990).
[37] K. A. Dill, K. M. Fiebig, and H. S. Chan, Proc. Natl. Acad. Sci. U.S.A. **90**, 1942 (1993).
[38] H. S. Chan and K. A. Dill, J. Chem. Phys. **95**, 3775 (1991).
[39] R. König and T. Dandekar, BioSystems **50**, 17 (1999).