Biased Monte Carlo optimization of protein sequences

Adrian P. Cootes, Paul M.G. Curmi, and Andrew E. Torda

Citation: The Journal of Chemical Physics **113**, 2489 (2000); doi: 10.1063/1.482067
View online: http://dx.doi.org/10.1063/1.482067
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/113/6?ver=pdfcov
Published by the AIP Publishing

Biased Monte Carlo optimization of protein sequences

Adrian P. Cootes
Research School of Chemistry, The Australian National University, Canberra ACT 0200, Australia

Paul M. G. Curmi
Initiative in Biomolecular Structure, School of Physics, The University of New South Wales, Sydney NSW 2052, Australia

Andrew E. Torda[a)]
Research School of Chemistry, The Australian National University, Canberra ACT 0200, Australia

(Received 26 January 2000; accepted 10 May 2000)

We demonstrate the application of a biased Monte Carlo method for the optimization of protein sequences. The concept of configurational-biased Monte Carlo has been used, but applied to sequence/composition rather than coordinates. Sequences of two-dimensional lattice proteins were optimized with the new approach and results compared with conventional Monte Carlo and a self-consistent mean-field (SCMF) method. Biased Monte Carlo (MC) was far more efficient than conventional MC, especially on more complex systems and with faster cooling rates. Biased MC did not converge as quickly as SCMF, but often found better sequences. © *2000 American Institute of Physics.* [S0021-9606(00)51030-7]

I. INTRODUCTION

If the amino acid sequence of a protein is written down, there is a very good chance a molecular biologist can produce it in useful quantities. Unfortunately, the ability to design a ''better'' amino acid sequence lags behind the experimental capability to produce it.[1] It remains remarkably difficult to find an approximation to an ideal protein sequence and it is only recently that there have been examples of large-scale protein redesign where one takes a given structure and attempts to find a sequence that will be more stable.[2] The practical applications are clear. It would often be useful to take a native protein and change the amino acid sequence to make it more heat stable or perhaps change it in part so as to accommodate some chemical modification.

There are two distinct aspects to the sequence design problem. First, there is the issue of how to best represent and calculate the compatibility of sequence and structure.[3] This requires a scoring function which may typically be based on physical principles,[4] knowledge-based approaches,[5] or a specifically designed function.[6] The second aspect is the search problem and is the subject of this study. Given some score or energy function, how can the optimum sequence be found?

The number of possible sequences grows very rapidly with protein size (20^N), but only a small number of these will be compatible with the structure of interest. The choice of search algorithm will depend on the computer time available and the type of answer desired. Sequence optimization is normally considered a discrete problem and this suggests certain optimization methods such as Monte Carlo (MC)[4,7,8] or genetic algorithms.[5,9] From the brute force point of view, a pruning algorithm known as the dead end elimination pruning algorithm[10] has also been used to design a small protein.[2] Recently, in an effort to try and avoid problems associated with large energy barriers and rugged search spaces, mean-field approaches have been receiving some interest.[11] This may be seen as an approach which by-passes the discrete nature of the problem (sites have partial amino acid character) and may also be promising for protein sequence optimization.[12–15]

MC has several attractive properties in principle, with practical disadvantages. With infinite computer time and slow cooling (simulated annealing) it will find the lowest energy sequence. It also has the desirable property that at finite temperature, it does not offer just one solution, but an ensemble of solutions with a known (Boltzmann) distribution. Since computer time is finite, it would be desirable to improve the sampling ability of the method while retaining its advantages.

When this problem is encountered in other fields, one approach is to introduce a bias in the selection of trial MC moves whose influence can be corrected by a more elaborate acceptance criterion (Rosenbluth) so as to maintain detailed balance and a Boltzmann distribution.[16] In this work, we introduce such a scheme for protein sequence optimization. By analogy with configuration-biased Monte Carlo (BMC),[17] the amino acid composition can be changed for several sites, guided by the local energy surface, and followed by application of the Rosenbluth acceptance criterion. BMC has been used previously in sequence design studies to efficiently generate decoy structures,[18] but not to actually optimize sequences.

The method has been tested on a very simple two-dimensional lattice model system which could be a protein or polymer. Calculations were run on systems of varying size and complexity (number of monomer types). For comparison, we have also implemented an SCMF method and conventional MC.

a)Electronic mail: Andrew.Torda@anu.edu.au

II. MATERIALS AND METHODS

A. The model

Compact proteins are represented as a self-avoiding walk of monomers on a fully occupied square, two-dimensional lattice. The sequence consists of the set of amino acids $\{\sigma_i\}$ where i is the position of the amino acid along the length of the chain. Each structure consists of a set of positions $\{r_i\}$, where each r_i is assigned to the monomer σ_i. The positions $\{r_i\}$ are defined so that the distance between consecutive monomers is one unit of the lattice and that no more than one monomer can exist at any one site. The structures surveyed were of length $N=16$, 36, and 64 monomers. For each length, 20 compact structures were generated randomly on 4×4, 6×6, and 8×8 lattices, respectively (shown in additional material[19]). Each structure within each set of 20 structures was unique and not related to any other within that set by rotational or translational symmetry.

The most common protein lattice representation may be the HP model,[20–22] where each monomer is either hydrophobic (H) or polar (P). Initial calculations suggested that a slightly more complicated model would better highlight the differences between methods. For this reason, 8 or 20 types of amino acids were used and monomers interacted with empty lattice sites. For convenience, this could be labeled a solvation, burial, or contact term, but since we are concerned with search methods, the physical interpretation is not relevant.

The energy (score) was given by

$$E_{\text{sequence}}=\sum_{i}^{N}\sum_{j>i}^{N}E^{\text{contact}}_{\sigma_i,\sigma_j}\Delta(r_i,r_j)+\sum_{i}^{N}E^{\text{contact\_number}}_{\sigma_i,\delta_i}, \tag{1}$$

where $E^{\text{contact}}_{\sigma_i,\delta_j}$ was the energy of contact between amino acid types σ_i and σ_j. The switching function $\Delta(r_i,r_j)=1$ if r_i and r_j were adjacent in the structure, but i and j not adjacent in sequence, and 0 otherwise. $E^{\text{contact\_number}}_{\sigma_i,\delta_i}$ is similar to a burial term used in many scoring functions.[23,24] δ_i is an index set to 0 (buried) if a site had zero or one empty adjacent lattice sites and set to 1 (exposed) otherwise. σ_i and δ_i were then used as indices to extract an energy $E^{\text{contact\_number}}_{\sigma_i,\delta_i}$ from the interaction matrix.

Given n amino acid/monomer types, all $n^2E^{\text{contact}}_{\sigma_i,\sigma_j}+2nE^{\text{contact\_number}}_{\sigma_i,\delta_i}$ interaction parameters were taken from a Gaussian distribution with an arbitrary mean and standard deviation of 0 and 1, respectively. This has the interesting property of giving asymmetric interactions ($E^{\text{contact}}_{\sigma_i,\sigma_j}\neq E^{\text{contact}}_{\sigma_j,\sigma_i}$), but has been proposed as a model for random protein sequences[25] and apparently mimics real protein interaction statistics.[26]

B. Optimization schemes

1. Biased Monte Carlo (BMC)

The BMC scheme used was based on configuration-biased Monte Carlo,[17] but using monomer type as the variable rather than configuration. A set of M random sites $\{l\}$ was selected for replacement, where the optimum value of M is system dependent and empirically determined. For the mth site, l_m, the Boltzmann weight of each amino acid type is given by

$$B_{l_m,\sigma_{l_m}}=e^{-(E_{l_m,\sigma_{l_m}}/kT)}, \tag{2}$$

where σ_{l_m} refers to the residue type being placed at site l_m. The energy E is calculated for the replaced residue in the field of the remaining sequence, including the previously selected amino acids (i.e., amino acids in the trial positions l_1 to $l_{\text{m-1}}$). The Boltzmann constant k was set to 1 for all calculations. At position l_m, the probability of each amino acid type σ_{l_m} is then calculated from

$$P_{l_m,\sigma_{l_m}}=\frac{B_{l_m,\sigma_{l_m}}}{\Sigma_{q=1}^{n}B_{l_m,q}}. \tag{3}$$

At each site, an amino acid was chosen randomly, but according to the probability $P_{l_m,\sigma_{l_m}}$ so as to introduce a bias to moves more likely to be accepted.

Therefore, the probability of generation of the trial sequence segment is given by

$$P=\prod_{m=1}^{M}P_{l_m,\sigma_{l_m}}. \tag{4}$$

The selection criterion, which corrects for the bias introduced in the sampling of sequences, compares the Rosenbluth weights of the trial and the original sequence. The Rosenbluth weight is given by

$$W=\prod_{m=1}^{M}\frac{1}{n}\sum_{q=1}^{n}B_{l_m,q}. \tag{5}$$

The criterion to be met for the acceptance of the newly generated sequence is

$$\xi\leqslant\frac{W_{\text{trial}}}{W_{\text{original}}}, \tag{6}$$

where ξ is a random number distributed uniformly between 0 and 1.

2. Self-consistent mean field

The energy E_{i,σ_i} of an amino acid of type σ_i at sequence position i in the weighted average field of all amino acids at all other positions in the structure is given by

$$E_{i,\sigma_i}=\sum_{\sigma_j=1}^{n}\sum_{j\neq i}^{N}E^{\text{contact}}_{\sigma_i,\sigma_j}\Delta(r_i,r_j)P^{\text{old}}_{j,\sigma_j}+E^{\text{contact\_number}}_{\sigma_i,\delta_i} \tag{7}$$

where $P^{\text{old}}_{j,\sigma_j}$ is the probability of the amino acid type σ_j occupying the position j from the calculation prior to the current calculation. The Boltzmann weight of an amino acid type σ_i at position i is then given by

$$B_{i,\sigma_i}=e^{-(E_{i,\sigma_i}/T)}. \tag{8}$$

The probability $P^{\text{new}}_{i,\sigma_i}$ of the amino acid type σ_i occupying the position i in the structure is then given by

$$P^{\text{new}}_{i,\sigma_i} = \frac{B_{i,\sigma_i}}{\sum_{q=1}^{n} B_{i,q}}. \tag{9}$$

The probability matrix P^{new} calculated from the previous matrix then gives the probability of occupation of each type of amino acid at each site in the protein. In the first step, the values for the probability matrix are taken randomly from a uniform distribution and then normalized so that the probabilities of each amino acid occurring at a given site sum to 1.

In order to suppress oscillations from the SCMF procedure, the new probability matrix P^{new} has a weighted contribution from the previous probability matrix P^{old} so that the new matrix $P^{\text{new}}_{\text{correct}}$ is given by

$$P^{\text{new}}_{\text{correct}} = \lambda P^{\text{old}} + (1-\lambda) P^{\text{new}}. \tag{10}$$

λ was set to a literature value[27] of 0.5, but also to 0.1 in some calculations as described in Sec. III.

3. Annealing schedule

For both MC/simulated annealing and SCMF calculations, the system was cooled by an exponential scheme where the temperature T at time t is given by

$$T(t) = T_0 e^{-(t/c)}, \tag{11}$$

where T_0 is the temperature at $t=0$ and c is a constant dictating the rate of cooling. Time t was taken as the processor time (proportional to the number of energy function evaluations) to enable direct comparison between the MC, BMC, and SCMF minimization algorithms.

The initial temperature was always set to $T_0 = 1000$ so that all sequences would be thermally accessible as the probabilities of least and most likely are within a few percent of each other. The cooling rates were labeled fast ($c=0.05$), medium ($c=0.1$), and slow ($c=1.0$). All minimization runs were terminated at $T=10^{-6}$. Since the difference in energy between the lowest states is of the order of 10^{-1} energy units, this means that the calculations were stopped well after the system was effectively frozen.

For each structure, 20 independent minimization runs were conducted from a random starting sequence and the average and standard deviation of the energy over these runs was calculated every 0.01 time unit.

C. Sequence entropies

Sequence entropies were calculated only on example 16-mer structures using constant temperature runs for both MC (step size of 1) and SCMF. In these calculations $T=0.2$, which was approximately the temperature at which the slowest optimization runs just converged. In MC, sufficient steps were taken to sample $\sim 10^6$ sequences and probabilities simply taken from the observed distributions. For the SCMF algorithm, the calculation was conducted as described previously (see Sec. II B 2) except that the temperature was held constant until the probability matrix converged. The convergence condition is given by

$$\sum_i \sum_{\sigma_i} (P^{\text{new}}_{i,\sigma_i} - P^{\text{old}}_{i,\sigma_i})^2 / Nn \leqslant 10^{-9}, \tag{12}$$

where $P^{\text{new}}_{i,\sigma_i}$ and $P^{\text{old}}_{i,\sigma_i}$ were the new and old probability matrices, N the number of monomer, and n the number of monomer types as above.

Given the probabilities P_{i,σ_i} of each amino acid type σ_i at position i, the sequence entropy S_i at position i is defined by

$$S_i = \sum_{\sigma_i=1}^{n} P_{i,\sigma_i} \ln P_{i,\sigma_i}. \tag{13}$$

III. RESULTS

The first calculations compared convergence properties of BMC, SCMF, and classic MC.

A. Dependence of MC and BMC algorithm on step size

BMC involves changing a whole segment of sequence as part of one trial move, but the size of the segment is not known in advance. Tests were performed with one, two, four, six, eight, or ten sites changed per trial. The results are shown in Fig. 1(a) for the largest system studied (64-mer) with the medium cooling rate. For the 20 compact structures tested, changing more than four sites per trial did not improve convergence speed, so this was used in subsequent calculations. Runs on smaller systems or different cooling rates showed the same trends. This result is almost certainly a reflection of the lattice model and the fact that monomers can never have more than three interacting neighbors.

It could be that conventional (unbiased) Monte Carlo would also benefit from moves that consist of more than one simultaneous change. This was tested by using moves that consisted of randomly changing more than one site before calculating the conventional Metropolis acceptance criterion. Figure 1(b) suggests that there is no advantage in changing more than one site at a time. Apparently, without any bias, any benefit from larger step sizes is outweighed by an increased rejection rate. All subsequent applications of the MC algorithm changed only one amino acid per optimization step.

B. Comparison of optimization methods

Sequence optimizations were conducted on structures of 16, 36, and 64 monomers using 20 maximally compact structures in each case and with three different cooling rates. In Fig. 2 (and subsequent Figs. 3, 5, 6, and 7), panels (a), (b), and (c) represent the results of fast, medium, and slow cooling, respectively.

Considering the smallest structures (16-mer), Fig. 2 shows the average of minimizations with 20 different structures. For each rate of cooling, the SCMF algorithm clearly converged to a low energy faster than either the MC or BMC methods, but there was little difference between the MC and BMC methods. All methods converged to sequences with almost identical energies by the time the temperature cooled to $T=10^{-6}$ for each rate of cooling (data not shown). Re-

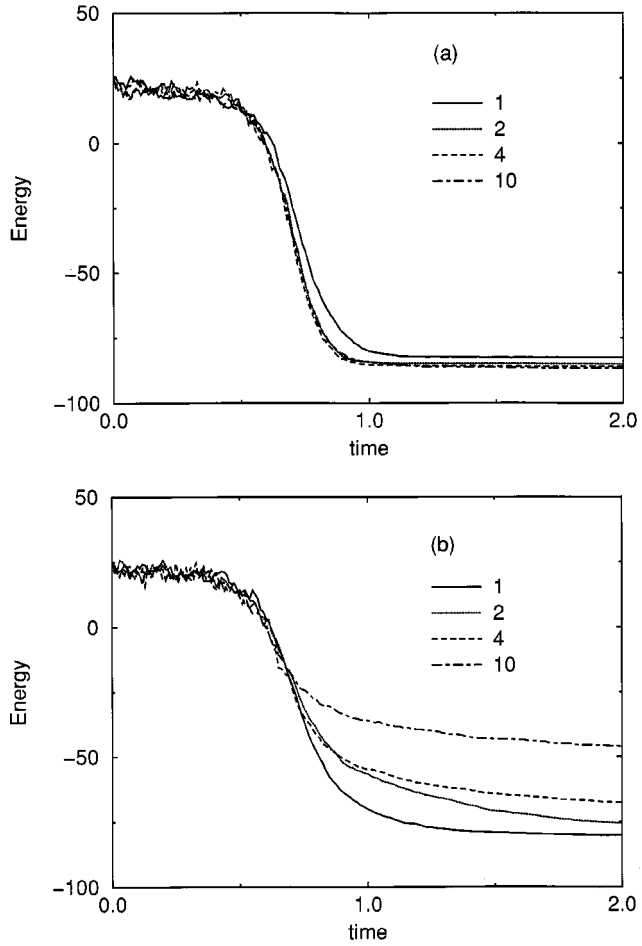

FIG. 1. Dependence of Monte Carlo optimization on move size. Trial moves consisted of one, two, four, or ten simultaneous residue changes as labeled for (a) BMC, and (b) MC. The structure was the first 64-mer listed in additional material and the cooling rate used is $c=0.1$.

peating the calculations for all of the 16-mer structures listed in Ref. 19 gave similar results (data not shown). Repeating the calculations for 36-mers suggested a difference between MC and BMC methods, but it was within statistical error.

For the larger 64-mer structures, the optimization speeds were, however, quite distinct. The calculations were done for all the structures listed in Ref. 19, but for clarity, Fig. 3 shows the results from the first structure as it was typical of all cases. The SCMF method minimizes fastest for all three cooling rates. For MC and BMC, there is a more interesting result. If cooling is slow enough [Fig. 3(c)], there is little difference. With faster cooling [Fig. 3(a)], BMC is much more efficient.

The example plots do not give any indication as to the statistical significance of the differences in convergence rates. One quick measure is to consider the standard deviations among the different calculations. Figure 4 shows the same runs as Fig. 2(b) (16-mer) and Fig. 3(b) (64-mer), but with error bars indicating the standard deviations among the energies. For the 16-mer, any difference between MC and BMC is not significant. For the larger 64-mer with medium or fast cooling rates, the difference between methods is much larger than the spread of results.

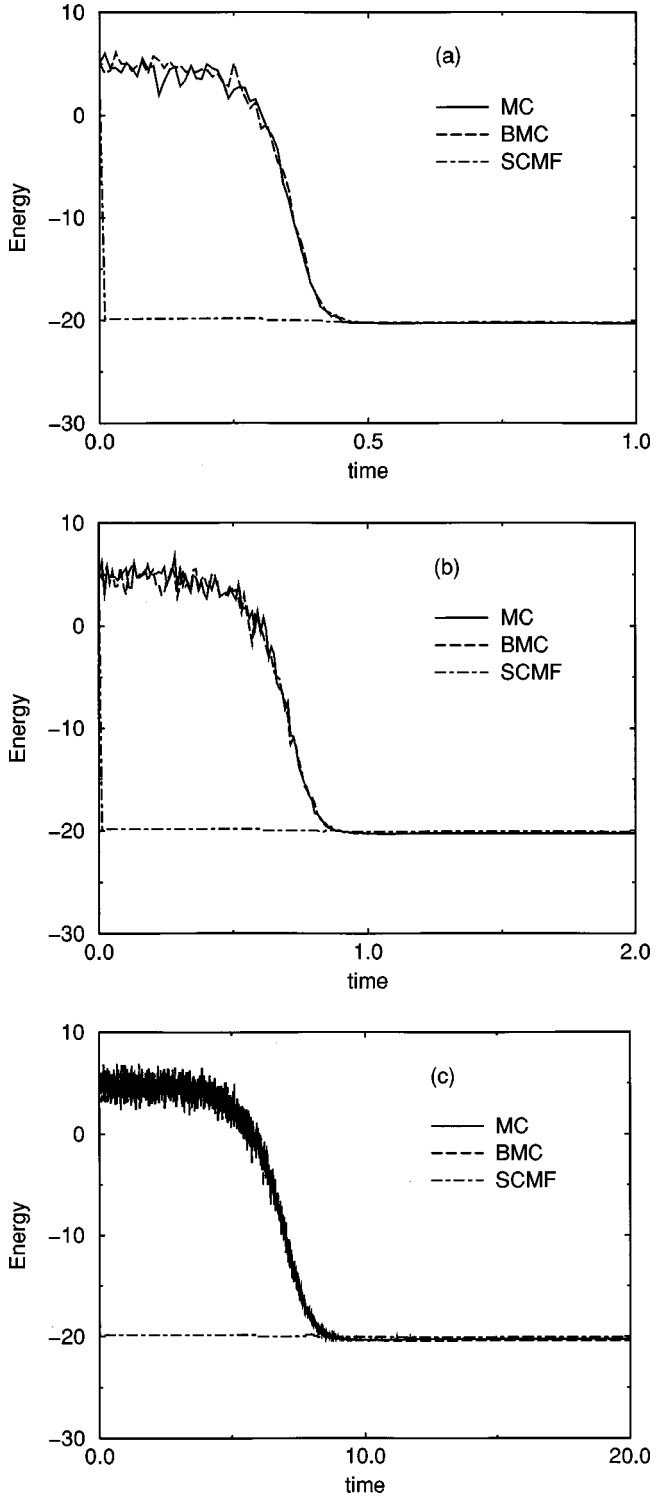

FIG. 2. Optimization of 16-mer. The average of 20 independent sequence energy minimizations is shown for MC, BMC, and SCMF for cooling rates $c=$ (a) 0.05, (b) 0.1, and (c) 1. The structure used is the first 16-mer listed in additional material (Ref. 19).

Based on the results so far, it would appear that SCMF is simply superior to Monte Carlo of any kind. While it is true that the convergence rate is much faster, there is a severe problem. SCMF does not always find solutions of energy as low as the Monte Carlo methods. This is shown in Fig. 5, where final energies are plotted for each 64-mer for the three

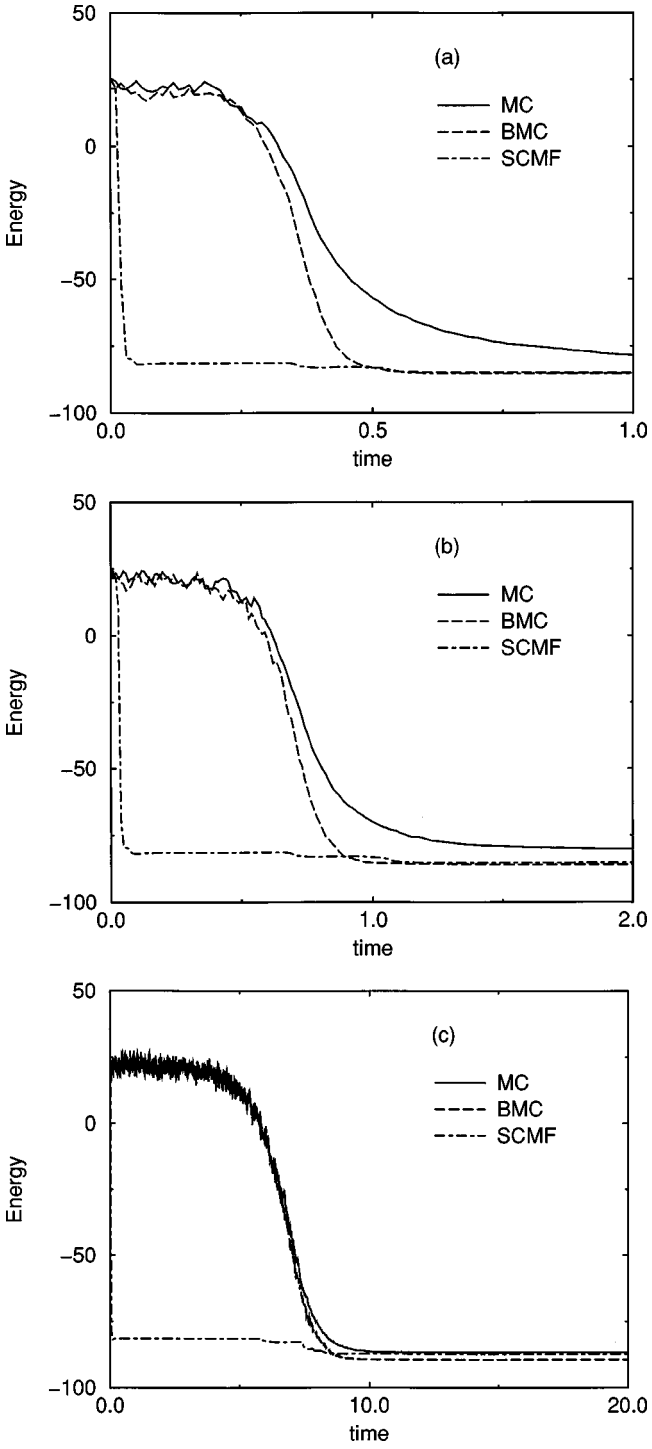

FIG. 3. Optimization of 64-mer. The average of 20 independent sequence energy minimizations is shown for MC, BMC, and SCMF for cooling rates $c=$ (a) 0.05, (b) 0.1, and (c) 1. The structure used is the first 64-mer listed in additional material (Ref. 19).

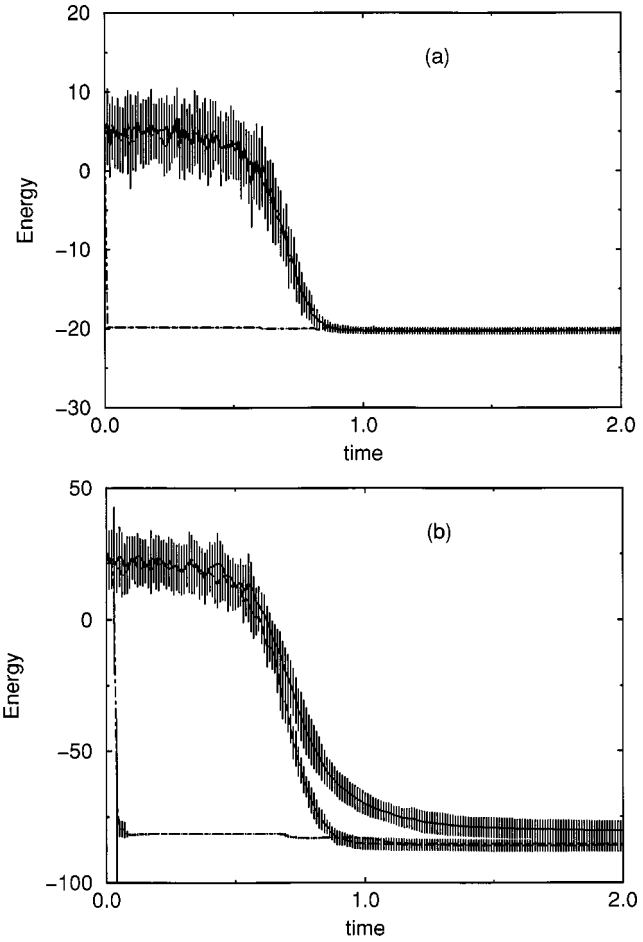

FIG. 4. Energy scatter during optimization. (a) corresponds to Fig. 2(b) and (b) to Fig. 3(b) except both have standard deviations of energies given as error bars.

rates of cooling and for the three different methods. Unless the cooling rate is very slow [Fig. 5(c)], classic MC does not find a good solution. For the middle cooling rate [Fig. 5(b)], BMC usually finds a better solution than SCMF, but the differences may not be significant compared with the standard deviations. For the slowest cooling rate [Fig. 5(c)], BMC always finds a significantly better result than SCMF. This can be interpreted in terms of final sequences. The mean interaction energy parameter was 0.7 units, which is of similar size to the difference in final energies between the two best methods. Although it is only an average value, it could be said that there is typically one better interaction in the BMC optimized sequences. While the BMC runs with the slowest cooling produce the best results, there is no proof that the results are the optimal solutions or that the system was in equilibrium in all calculations. Most importantly, different runs do not converge to the same solution. Apparently the cooling rate is in a regime where it will produce results that are better than SCMF, but still not perfect.

C. Dependence on number of amino acids and SCMF damping factor

All the calculations described above used eight amino acid types, but it is interesting to see if the results change when the system is made much more complex. For this reason, a few calculations were done with 20 amino acid/monomer types. This increases the size of the search space, so it becomes difficult for any method to approach the global minimum.

Figure 6 shows the results for three cooling rates for the first 64-mer structure listed in additional[19] material with each method. For the slowest cooling rates [Fig. 6(c)], the results

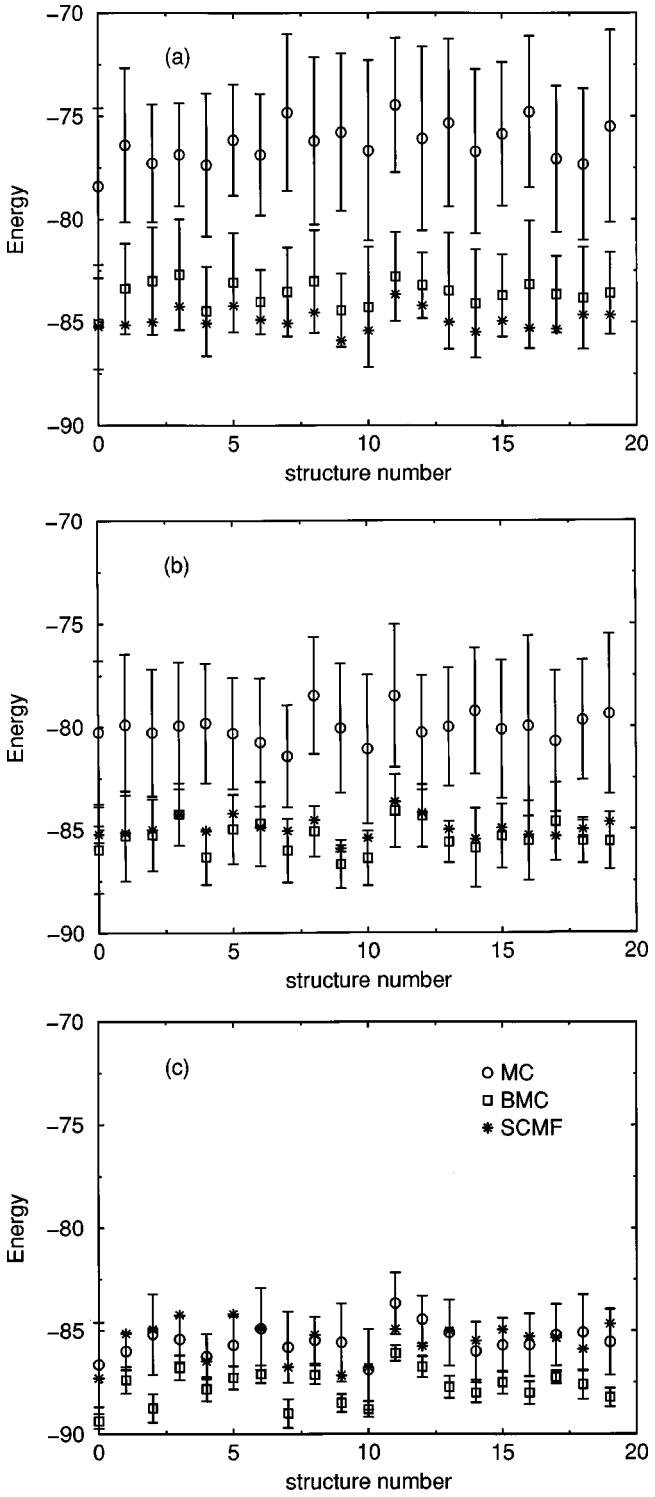

FIG. 5. Final energies for 64-mer structures with eight monomer types. The average and standard deviations of final energies are shown for each structure. Structure number corresponds to the order structures are listed in additional material. Cooling rates are $c=$ (a) 0.05, (b) 0.1 and (c) 1.

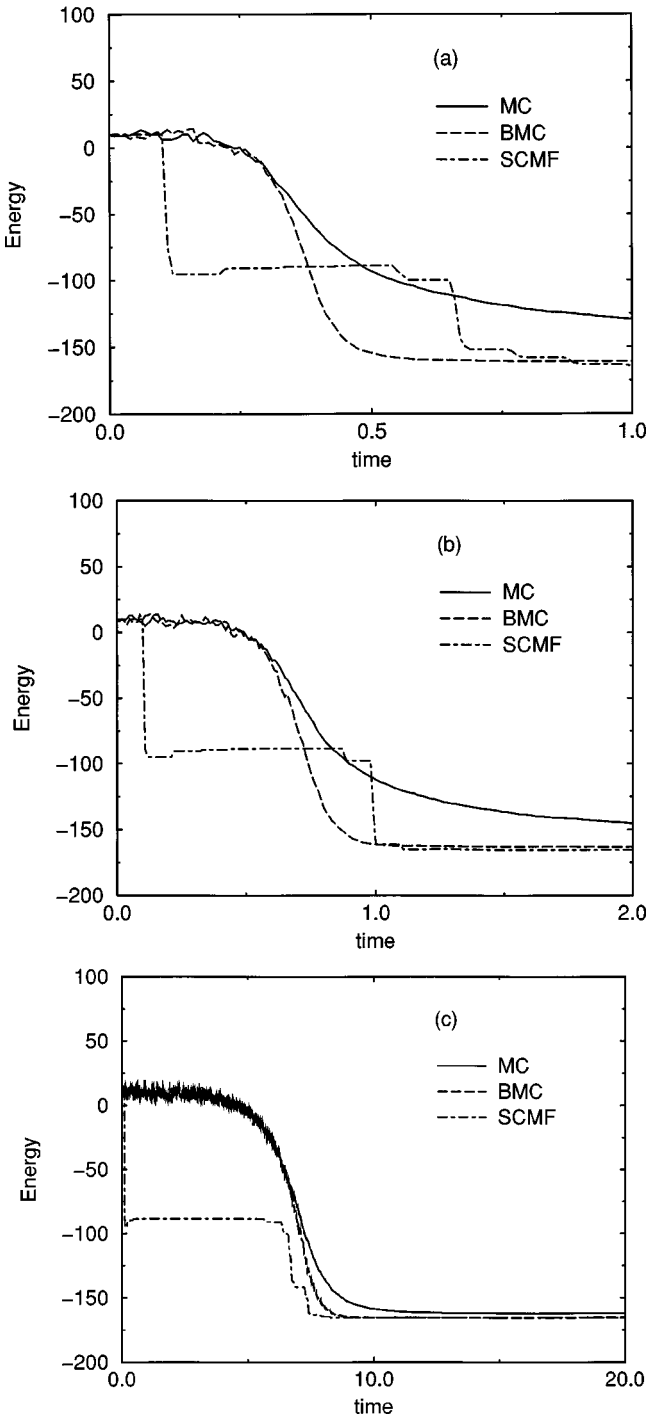

FIG. 6. Sequence optimization of 64-mer with 20 amino acid types. Labels and symbols as per Fig. 2.

are not surprising, suggesting that SCMF finds a solution most rapidly. For the faster cooling rates [Fig. 6(a)], however, there is a different result. For these cooling speeds, BMC reaches a very good solution faster than SCMF. This was not seen in the previous calculations with eight monomer types.

These results, however, show more interesting behavior for SCMF. There are parts of the calculation where the energy value seems to plateau. This led to the question as to whether the parameters were best adapted to the system. In fact, SCMF does contain one very arbitrary parameter, the λ (damping) value given in Eq. (10). It is essential that λ be positive and nonzero to prevent oscillations. The value of $\lambda = 0.5$, however, was simply taken from the literature. The calculations with 20 amino acid types were then repeated, but after setting $\lambda = 0.1$; the results are shown in Fig. 7. With this value of λ, SCMF generally appears to be faster con-

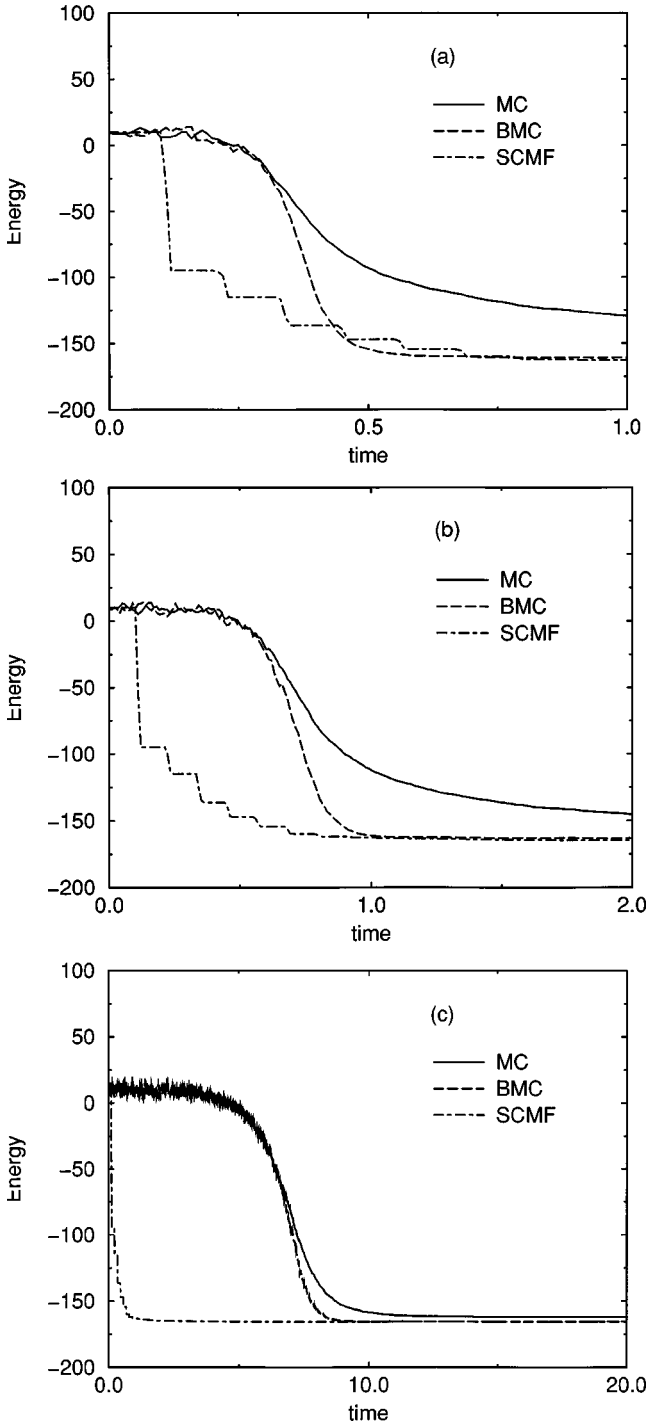

FIG. 7. Effect of λ on SCMF optimization. Labels and symbols as per Fig. 6, but with $\lambda = 0.1$.

verging and no oscillations in the probability matrix were observed. These results highlight the fact that parameter settings can often be set arbitrarily and will affect any comparison of methods.

IV. DISCUSSION

Classic Monte Carlo has a history of application to sequence optimization,[4,7] but there appears to be no reason to use it. The biased Monte Carlo method introduced here is consistently and reproducibly more efficient while maintaining the same statistical mechanical properties. This may not be surprising given the success of analogous methods such as configuration-biased Monte Carlo.[17] Furthermore, it appears that this is simply due to the difference in the sampling/acceptance method. Even when a classic Monte Carlo step involved changing more than one residue at a time, MC was not a competitive approach.

Beyond this clear difference, the method of choice depends on the system and the goals. If a sampling of sequences with a known (Boltzmann) distribution is wanted, BMC may be the only useful method. It does, however, usually perform more slowly than SCMF. SCMF, however, has the distinct disadvantage that the answers, which are quickly produced, are not always correct (it is not guaranteed to converge to the correct sequence). Of course, in this study, sequences were optimized with relatively rapid cooling schedules and it remains possible that SCMF would find the correct sequence for these systems given a slower cooling regime. In practice, applying SCMF with rapid cooling may not be a problem since the quality of the results was never far from BMC and the error due to lack of convergence may be smaller than the error due to force field/score function approximations. SCMF also has a less obvious disadvantage. Any fast implementation relies on a large matrix of stored interactions. With n monomer types and structures of length N, this grows with n^2N^2.

The work here has not dwelt on the calculation of sequence entropies or sequence information content [Eq. (13)]. Its physical meaning is debatable, but it may be seen as a measure of how much a particular site is allowed to vary. It is readily accessible from the distributions in an equilibrium BMC simulation or the probability matrix of an SCMF calculation. For the calculation of sequence entropy, similar considerations apply as to a minimization. This measure is a property of the neighborhood being sampled. If it is the wrong neighborhood, the results may suffer correspondingly. Figure 8 gives examples of sequence entropies derived from MC and SCMF simulations. Each diagram shows a structure and at each site, the bars show the sequence entropy calculated by each method. In Fig. 8(a), there is good agreement, but Fig. 8(b) shows an example where SCMF converged to a worse sequence. In this case, the larger/smaller bars show where it has over-underestimated the sequence entropy.

After using a simple model system, the question of transferability to more realistic systems is always posed. Obviously, no specific parameters would be transferable to a more complicated protein-like system with continuous (nonlattice) coordinates, 20 amino acid types, and more intricate interaction functions. It is also clear that this kind of approach is best suited to coarse-grained, low-resolution (nonatomistic) models. Possibly the most drastic change with a more realistic protein would be the number of sites changed per trial move. In a real protein, each site has many more neighbors and longer range interactions. Aside from specific parameters, some trends would certainly be transferable to a more realistic protein model. The advantages of biased Monte Carlo moves over simple MC are clear. The potential disadvantages of SCMF are also clear. The possibility of convergence to an incorrect minimum will only increase as systems

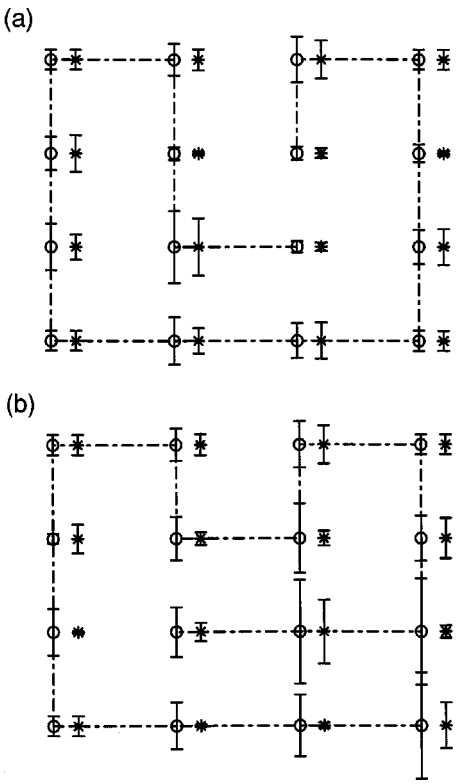

FIG. 8. Sequence entropies for 16-mers calculated with MC and SCMF methods. Dotted lines represent the lattice structure and the error bars represent the sequence entropy at that point in the structure. The sizes of the error bars are proportional to the value of the sequence entropy. Large entropies indicate a point with high mutability; small entropies have low mutability. Entropy calculations were conducted at $T=0.2$; (a) is structure no. 10 and (b) is structure no. 5 [as listed in the additional material (Ref. 19)].

are made more complicated. Furthermore, results on even a simple system highlight the importance of parameters such as the λ (damping) parameter in SCMF and this would also have to be tuned to any real system.

Finally, the work here has compared methods that have well-understood distribution properties and has not touched on methods that are purely optimization tools such as the dead end elimination algorithm and genetic algorithm. If ensemble properties are not of interest, these may be appropriate devices.

ACKNOWLEDGMENT

We thank the mighty Thomas Huber for supplying the code for the generation of lattice structures.

[1] D. T. Jones, Curr. Opin. Biotech. **6**, 452 (1995).
[2] B. I. Dahiyat and S. L. Mayo, Science **278**, 82 (1997).
[3] A. G. Street and S. L. Mayo, Structure **7**, R105 (1999).
[4] H. W. Hellinga and F. M. Richards, Proc. Natl. Acad. Sci. USA **91**, 5803 (1994).
[5] D. T. Jones, Protein Sci. **3**, 567 (1994).
[6] T-L. Chiu and R. A. Goldstein, Protein Eng. **11**, 749 (1998).
[7] J. R. Desjarlais and N. D. Clarke, Curr. Opin. Struct. Biol. **8**, 471 (1998).
[8] E. I. Shakhnovich and A. M. Gutin, Protein Eng. **6**, 793 (1993).
[9] M. Ebeling and W. Nadler, Biopolymers **41**, 165 (1997).
[10] J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters, Nature (London) **356**, 539 (1992).
[11] P. Koehl and M. Delarue, Curr. Opin. Struct. Biol. **6**, 222 (1996).
[12] P. Koehl and M. Delarue, J. Chem. Phys. **108**, 9540 (1998).
[13] H. Kono and J. Doi, Proteins **19**, 244 (1994).
[14] H. Kono, M. Nishiyama, M. Tanokura, and J. Doi, Pac. Symp. Biocomput. **2**, 210 (1997).
[15] J. G. Saven and P. G. Wolynes, J. Phys. Chem. B **101**, 8375 (1997).
[16] M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).
[17] J. I. Siepmann, in *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, edited by W. F. Gunsteren, P. K. Weiner, and A. J. Wilkinson (ESCOM, Leiden, 1993), pp. 249–264.
[18] F. Seno, M. Vendruscolo, A. Maritan, and J. R. Banavar, Phys. Rev. Lett. **77**, 1901 (1996).
[19] http://www.rsc.anu.edu.au/~torda/papers/lattice.html
[20] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan, Protein Sci. **4**, 561 (1995).
[21] E. I. Shakhnovich, Folding Des. **3**, R45 (1998).
[22] K. F. Lau and K. A. Dill, Macromolecules **22**, 3986 (1989).
[23] M-H. Hao and H. A. Scheraga, Physica A **244**, 124 (1997).
[24] S. Premilat and O. Collet, Europhys. Lett. **39**, 575 (1997).
[25] A. Sali, E. Shakhnovich, and M. Karplus, J. Mol. Biol. **235**, 1614 (1994).
[26] A. P. Cootes, P. M. G. Curmi, R. Cunningham, C. Donnelly, and A. E. Torda, Proteins **32**, 175 (1998).
[27] P. Koehl and M. Delarue, J. Mol. Biol. **239**, 249 (1994).