Growth-based optimization algorithm for lattice heteropolymers

Hsiao-Ping Hsu, Vishal Mehra, Walter Nadler, and Peter Grassberger
John-von-Neumann Institute for Computing, Forschungszentrum Jülich, D-52425 Jülich, Germany
(Received 16 September 2002; published 27 August 2003)

An improved version of the pruned-enriched-Rosenbluth method (PERM) is proposed and tested on finding lowest energy states in simple models of lattice heteropolymers. It is found to outperform not only the previous version of PERM, but also all other fully blind general purpose stochastic algorithms which have been employed on this problem. In many cases, it found new lowest energy states missed in previous papers. Limitations are discussed.

DOI: 10.1103/PhysRevE.68.021113 PACS number(s): 05.10.−a, 87.15.By, 87.10.+e

Lattice polymers have been studied intensively to understand protein folding, one of the central problems of computational biology. A popular model used in these studies is the so-called HP model [1,2] where only two types of monomers, *H* (hydrophobic) and *P* (polar) ones, are considered. Hydrophobic monomers tend to avoid water which they can do only by mutually attracting themselves. The polymer is modeled as a self-avoiding chain on a regular (square or simple cubic) lattice with interactions $(\epsilon_{HH},\epsilon_{HP},\epsilon_{PP})=-(1,0,0)$ between neighboring nonbonded monomers.

This model might be too simple to represent finer details of real protein folding [3], but this is not our concern. We use the search for its ground states as a paradigmatic example for combinatorial optimization, with a large body of existing benchmarks.

A wide variety of computational strategies have been employed to simulate and analyze these models, including conventional (Metropolis) Monte Carlo schemes with various types of moves [4–6], chain growth algorithms without [7] and with resampling [8–10] (see also [11]), genetic algorithms [12,13], parallel tempering [14] and generalizations thereof [15,16], an ''evolutionary Monte Carlo'' algorithm [17], and others [18]. In addition, Yue and Dill [19] also devised an exact branch-and-bound algorithm specific for HP sequences on cubic lattices, which gives all low energy states by exact enumeration and typically works for $N\lesssim 70-80$.

It is the purpose of the present paper to present an improved variant of the pruned-enriched Rosenbluth method (PERM) [20] and to apply it to lattice proteins. PERM is a biased chain growth algorithm with resampling (''population control'') and depth-first implementation. It is built on the old idea of Rosenbluth and Rosenbluth [21] to use a biased growth algorithm for polymers, where the bias is corrected by means of giving a weight to each sample configuration. While the chain grows by adding monomers, this weight (which also includes the Boltzmann weight if the system is thermal) will fluctuate. PERM suppresses these fluctuations by ''pruning'' configurations with too low weight and by ''enriching'' the sample with copies of high-weight configurations [20]. These copies are made while the chain is growing, and continue to grow independently of each other. PERM can be viewed as a special realization of a ''go-with-the-winners'' strategy [22] and indeed dates back to the beginning of the Monte Carlo simulation era, when it was called ''Russian roulette and splitting'' [23]. Among statisticians, this approach is also known as sequential importance sampling with resampling [24].

Pruning and enrichment are done by choosing thresholds $W_n^<$ and $W_n^>$ depending on the estimate of the partition sums of *n*-monomer chains (see below for their actual determination). If the current weight W_n of an *n*-monomer chain is less than $W_n^<$, the chain is discarded with probability 1/2, otherwise it is kept and its weight is doubled. Many alternatives to this simple choice are discussed in Ref. [24], but we found that more sophisticated strategies had little influence on the efficiency, and thus we kept the above in the present work. On the contrary, we found that different strategies in biasing and, most of all, in enrichment had a big effect, and it is here where the present variant differs from those in Refs. [8,9]. There, high-weight configurations were simply cloned and the weight was uniformly shared between the clones. For relatively high temperatures this is very efficient [20], since each clone has so many possibilities to continue that different clones very quickly become independent from each other. This is no longer the case for very low temperatures. There we found that clones often evolved in the same direction, since one continuation has a much higher Boltzmann weight than all others. Thus, cloning is no longer efficient in creating configurational diversity, which was the main reason why it was introduced.

The main modification made in the present paper is thus that we no longer make *identical clones*. Rather, when we have a configuration with $n-1$ monomers, we first estimate a *predicted* weight $W_n^{\rm pred}$ for the next step, and we count the number $k_{\rm free}$ of free sites where the *n*th monomer can be placed. If $k_{\rm free}>1$ and $W_n^{\rm pred}>W_n^>$, we choose $2\leqslant k\leqslant k_{\rm free}$ *different* sites among the free ones and continue with *k* configurations which are *forced* to be different. Thus, we avoid the loss of diversity which limited the success of old PERM. Typically, we used $k=\min\{k_{\rm free},\lceil W_n^{\rm pred}/W_n^>\rceil\}$.

When selecting a *k* tuple $A=\{\alpha_1,\ldots,\alpha_k\}$ of mutually different continuations α_j with probability p_A, the corresponding weights $W_{n,\alpha_1},\ldots,W_{n,\alpha_k}$ are

$$W_{n,\alpha_j}=\frac{W_{n-1}q_{\alpha_j}k_{\rm free}}{k\binom{k_{\rm free}}{k}p_A}, \tag{1}$$

where the *importance* $q_{\alpha_j}=\exp(-\beta E_{n,\alpha_j})$ of choice α_j is the

Boltzmann-Gibbs factor associated with the energy E_{n,α_j} of the newly placed monomer in the potential created by all previous monomers. The other terms arise from correcting bias and normalization, see Ref. [25] for a more thorough discussion. Choosing

$$p_A = \frac{\sum_{\alpha \in A} q_\alpha}{\sum_{A'} \sum_{\alpha' \in A'} q_{\alpha'}} \tag{2}$$

would result in usual importance sampling [25]. However, instead of q_α we use the modified importances $\tilde{q}_\alpha = (k_{\text{free}}^{(\alpha)} + 1/2) q_\alpha$ in Eq. (2), $k_{\text{free}}^{(\alpha)}$ being the number of free neighbors when the nth monomer is placed at α. This replacement is made since we anticipate that continuations with less free neighbors will contribute less on the long run than continuations with more free neighbors. This is similar to ''Markovian anticipation'' [26] within the framework of old PERM, where a bias different from the short-sighted optimal importance sampling was found to be preferable. Consequently, the predicted weight is $W_n^{\text{pred}} = W_{n-1} \Sigma_\alpha \tilde{q}_\alpha$,

A noteworthy feature of new PERM is that it crosses over to complete enumeration when $W_n^<$ and $W_n^>$ tend to zero. In this limit, all possible branches are followed and none is pruned as long as its weight is not strictly zero. In contrast to this, old PERM would have made infinitely many copies of the same configuration. This suggests already that we can be more lenient in choosing $W_n^<$ and $W_n^>$. For the first configuration hitting length n, we used $W_n^< = 0$ and $W_n^> = \infty$, i.e., we neither pruned nor branched. For the following configurations, we used $W_n^> = Z_n / Z_0 (c_n / c_0)^2$ and $W_n^< = 0.2\, W_n^>$. Here, c_n is the total number of configurations of length n already created during the run and Z_n is the partition sum estimated from these configurations.

In PERM, we work at a fixed temperature (no annealing), and successive ''tours'' [20] are independent except for the thresholds $W_n^{<,>}$ which use partially the same partition sum estimates. Results are less sensitive to the precise choice of temperature than they were for old PERM. In general, all temperatures in the range $0.25 < T < 0.35$ gave good results for ground state search. In the following, when we quote numbers of ground state hits or CPU times between such hits, these are always *independent* hits. The actual numbers of (dependent) hits are much larger.

We now present our results. Special comparison is made with the *core-directed growth method* (CG) of Beutler and Dill [11], the only method we found to be still competitive with ours. We emphasize, however, that the CG method works only for the HP model and relies heavily on heuristics, in contrast to our fully blind general purpose approach.

(a) We first tested the ten 48-mers from Ref. [4]. As with old PERM, we could reach lowest energy states for all of them, but within much shorter CPU times. For all ten chains, we used the same temperature, $\exp(1/T) = 18$, although we could have optimized CPU times by using different temperatures for each chain.

TABLE I. Performances for the 3D Binary (HP) sequences from Ref. [4].

Sequence No.	$-E_{\min}$[a]	PERM[b]	New PERM[c]	New PERM[d] with bias [27]
1	32	6.9	0.63	0.13
2	34	40.5	3.89	0.23
3	34	100.2	1.99	0.71
4	33	284.0	13.45	6.57
5	32	74.7	5.08	2.55
6	32	59.2	6.60	1.44
7	32	144.7	5.37	3.35
8	31	26.6	2.17	0.46
9	34	1420.0	41.41	10.53
10	33	18.3	0.47	0.08

[a]Ground state energies [4].
[b]CPU times (minutes) per independent ground state hit, on 167-MHz Sun ULTRA I work station; from Ref. [9].
[c]CPU times (minutes), same machine
[d]CPU times (minutes), same machine.

The CPU times for new PERM in Table I are typically one order of magnitude smaller than those in Ref. [11], except for sequence No. 9 whose lowest energy was not hit in Ref. [11]. Since in Ref. [11] a SPARC 1 machine was used which is slower by a factor of ≈ 10 than the 167-MHz Sun ULTRA I used here, this means that our algorithms have comparable speeds. We note that introducing a simple configurational bias in new PERM [27] can already give a considerable speed up; in this contribution, however, we want to concentrate on blind search.

(b) Next we studied the two 2D (two-dimensional) HP sequences of length $N = 100$ of Ref. [5]. They were originally thought to have ground states fitting into a 10×10 square with energies -44 and -46 [5], but in Ref. [9] configurations fitting into this square were found with lower energies. Moreover, when configurations were allowed to have arbitrary shape, even lower energies were found [9,10,15]. In the present work, we studied only configurations of the latter type. The lowest energies known by now are -48 [10] resp. -50 [15]. The CPU times needed to find them were 48 min respectively 50 h, on machines with ≈ 500 MHz. In contrast, new PERM needed on average 2.6 min respectively 5.8 h on a 667-MHz DEC Alpha 21264 between any two hits.

(c) Several 2D HP sequences were introduced in Ref. [12], where the authors tried to fold them using a genetic algorithm. Except for the shortest chains they were not successful, but putative ground states for all of them were found in Refs. [9,14,15]. But for the longest of these chains ($N = 64$), the ground state energy $E_{\min} = -42$ was found in Ref. [9] only by means of special tricks which amount to nonblind search. With blind search, the lowest energy reached by PERM was -39. We should stress that PERM as used in Ref. [9] was blind for all cases except this 64-mer (and when it found $E = -49$ for the second $N = 100$ chain of Ref. [5]), in contrast to statements to the contrary made in Ref. [17].

TABLE II. Newly found lowest energy states for binary sequences with interactions $\vec{\epsilon}=(\epsilon_{HH},\epsilon_{HP},\epsilon_{PP})=-(1,0,0)$.

N	d	Sequence *example conformation*[b]	Old $E_{\min}$ [Reference]	New $E_{\min}$	$e^{1/T}$	CPU time[a]
85	2	$H_4P_4H_{12}P_6H_{12}P_3H_{12}P_3H_{12}P_3HP_2H_2P_2H_2P_2HPH$ $flb_3lf_4lf_2rbrbrfr_2f_3l_2b_2lf_2lbl_2frfl_2b_2lbr_2b_3rb_3l_2frflf_3lb_5lf_2lfrflfrflfrfr$	-52 [17]	-53	90	0.03
58	3	$PHPH_3PH_3P_2H_2PHPH_2PH_3PHPHPH_2P_2H_3P_2HPHP_4HP_2HP_2H_2P_2HP_2H$ $ublfl_2urfldrfrbrub_2lf_3lublbrurdfrubdblbufldblfldr_2bdfdlu$	-42 [18]	-44	30	0.19
103	3	$P_2H_2P_5H_2P_2H_2PHP_2HP_7HP_3H_2PH_2P_6HP_2HPHP_2$ $HP_5H_3P_4H_2PH_2P_5H_2P_4H_4PHP_8H_5P_2HP_2$ $ufrbdflfurdfu_2rd_2buruf_2ulbluld_2burdrubrdl_2bufldblfulf_2rd$ $bd_2b_2uflufd_3fururd_2fu_2ru_2ldf_2urbl_2dbdlbulfru_2$	-49 [18]	-54 [27]	60	3.12
124	3	$P_3H_3PHP_4HP_5H_2P_4H_2P_2H_2P_4HP_4HP_2HP_2H_2P_3H_2PHPH_3P_4H_3P_6$ $H_2P_2HP_2HPHP_2HP_7HP_2H_3P_4HP_3H_5P_4H_2PHPHPHPH$ $urbd_2bublfurb_2drf_5ub_4ufluldfrufrbdfrbubd_2burf_3dlbrb_3df_2$ $lf_3urdb_2d_2luflb_2rbrfdrfrubulbuf_2u_2b_2dfrbdf_2dldf_2u_2bdrurbulfl$	-58 [18]	-71	90	12.3
136	3	$HP_5HP_4HPH_2PH_2P_4HPH_3P_4HPHPH_4P_{11}HP_2HP_3HPH_2P_3H_2P_2HP_2$ $HPHPHP_8HP_3H_6P_3H_2P_2H_3P_3H_2PH_5P_9HP_4HPHP_4$ $u_2b_2rdl_2frbdrdlf_2lfr_2brblu_3fd_2rbubd_2r_2df_3dl_2ul_2blbrdfrurbldrbul_2b$ $rdrurf_2urf_2ububdlu_2bd_2blurul_2d_2ldr_4ubld_2l_2urubu_3brd_2f_2u_2ld_2ldbubu$	-65 [18]	-80	120	110

[a]Hours per independent hit on 667-MHz DEC ALPHA 21264.
[b]r=right, l=left, f=forward, b=backward, u=up, d=down.

We now found putative ground states for all chains of Ref. [12] with blind search. For the 64-mer, the average CPU time per hit was ca. 30 h on the DEC 21264, which seems to be roughly comparable to the CPU times needed in Refs. [14,15], but considerably slower than Ref. [11]. This sequence is particularly difficult for any growth algorithm, and the fact that we now found it is particularly noteworthy.

On the other hand, new PERM was much faster than Ref. [11] for the sequence with $N=60$ of [12]. It needed ≈ 10 s on the DEC 21264 to hit $E_{\min}=-36$ and ≈ 0.1 s to hit $E=-35$. In contrast, $E=-36$ was never hit in Ref. [11], while it took 97 min to hit $E=-35$.

(d) An 85-mer 2D HP sequence was given in Ref. [28], where it was claimed to have $E_{\min}=-52$. Using a genetic algorithm, the authors could find only conformations with $E\geqslant -47$. In Ref. [17], using a newly developed *evolutionary Monte Carlo* method, the authors found the putative ground state when assuming large parts of its known structure as constraints. This amounts, of course, to nonblind search. Without these constraints, the putative ground state was not hit in Ref. [17] either, although the authors claimed their algorithm to be more efficient than all previous ones. We easily found states with $E=-52$, but we also found many conformations with $E=-53$. At $\exp(1/T)=90$, it took ca. 10 min CPU time between successive hits on the Sun ULTRA 1.

(e) Four 3D HP sequences with $N=58$, 103, 124, and 136 were proposed in Refs. [29,30] as models for actual proteins or protein fragments. Low energy states for these sequences were searched in Ref. [18] using a newly developed and supposedly very efficient algorithm. The energies reached in Ref. [18] were $E=-42$, -49, -58, and -65, respectively. We now found lower energy states after only few minutes of CPU time, for all four chains. For the longer ones, the true ground state energies are indeed *much* lower than those found in Ref. [18], see Table II.

Note the very low temperatures needed to fold the very longest chains in an optimal time. If we would be interested in excited states, higher temperatures would be better. For instance, to find $E=-66$ for the 136-mer (which is one unit below the lowest energy reached in Ref. [18]), it took just 2.7 s/hit on the DEC 21264 when using $\exp(1/T)=40$.

(f) The only case where we could not find a known ground state is a 3D HP sequence of length 88 given in Ref. [11]. As shown there, it folds into an irregular β/α barrel with $E_{\min}=-72$. The difficulties of PERM with this sequence are easily understood by looking at the configuration shown in Ref. [11]. The nucleus of the hydrophobic core is formed by amino acids Nos. 36–53. Before its formation, a growth algorithm starting at either end has to form very unstable and seemingly unnatural structures which are stabilized only by this nucleus, a situation similar to the 64-mer of Ref. [12]. In order to fold also this chain, we would have either to start from the middle of the chain (as done in Ref. [9] for some sequences) or use some other heuristics which help the formation of the hydrophobic core. Since we wanted our algorithm to be as general and ''blind'' as possible, we did not incorporate such tricks [27].

A more detailed discussion of our algorithm, the results, and comparison with other methods is given elsewhere [25]. A list containing all sequences for which we found improved lowest energy configurations is given in Table II.

In the present paper we presented an improved version of PERM which is a depth-first implementation of the ''go-with-the-winners'' strategy (or sequential importance sampling with resampling). The main improvement over old

PERM is that we now do not make *identical clones* of high-weight (partial) configurations, but we branch such that each continuation is forced to be different. We do not expect this to have much influence for systems at high temperatures, but as we showed, it leads to substantial improvement at very low temperatures.

Comparing our results to previous work, we see that we found the known lowest energy states in *all* cases but one. Moreover, whenever we could compare with previous CPU times, the comparison was favorable for our improved algorithms, except for the CG method of Beutler and Dill [11]. But we should stress that the latter is very specific to HP chains, uses strong heuristics regarding the formation of a hydrophobic core, and does not give correct Boltzmann weights for excited states. All that is not true for our method.

Although our method could be used for a much wider range of applications (see Ref. [31] for applications of PERM), we presented here only results for heteropolymers with two types of monomers and the simplest nontrivial interactions on the square and simple cubic lattices. But we applied it also successfully to the HP model on the FCC lattice, to off-lattice heteropolymers, and to lattice models with more than two types of monomers (to be published). We hope that our results will also foster applications to more realistic protein models. We showed only results for lowest energy configurations, but we should stress that PERM is not only an optimization algorithm. It also gives information on the full thermodynamic behavior. We skipped this here since finding ground states is the most difficult problem, in general, and sampling excited states is easy compared to it.

[1] K.A. Dill, Biochemistry **24**, 1501 (1985).

[2] K.F. Lau and K.A. Dill, Macromolecules **22**, 3986 (1989); H.S. Chan and K. Dill, J. Chem. Phys. **95**, 3775 (1991); D. Shortle, H.S. Chan, and K.A. Dill, Protein Sci. **1**, 201 (1992).

[3] For example, HP models cannot explain calorimetric cooperativity: H.S. Chan, Proteins **40**, 543 (2000).

[4] K. Yue *et al.*, Proc. Natl. Acad. Sci. U.S.A. **92**, 325 (1995).

[5] R. Ramakrishnan, B. Ramachandran, and J.F. Pekny, J. Chem. Phys. **106**, 2418 (1997).

[6] J.M. Deutsch, J. Chem. Phys. **106**, 8849 (1997).

[7] E.M. O'Toole and A.Z. Panagiotopoulos, J. Chem. Phys. **97**, 8644 (1992).

[8] H. Frauenkron, U. Bastolla, E. Gerstner, P. Grassberger, and W. Nadler, Phys. Rev. Lett. **80**, 3149 (1998).

[9] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, Proteins **32**, 52 (1998).

[10] J.L. Zhang and J.S. Liu, J. Chem. Phys. **117**, 3492 (2002).

[11] T.C. Beutler and K.A. Dill, Protein Sci. **5**, 2037 (1996).

[12] R. Unger and J. Moult, J. Mol. Biol. **231**, 75 (1993).

[13] R. König and T. Dandekar, Protein Eng. **14**, 329 (2001).

[14] A. Irbäck, in *Monte Carlo Approach to Biopolymers and Protein Folding*, edited by P. Grassberger *et al.* (World Scientific, Singapore, 1998), pp. 98–109.

[15] G. Chikenji, M. Kikuchi, and Y. Iba, Phys. Rev. Lett. **83**, 1886 (1999).

[16] G. Chikenji and M. Kikuchi, Proc. Natl. Acad. Sci. U.S.A. **97**, 14 273 (2000).

[17] F. Liang and W.H. Wong, J. Chem. Phys. **115**, 3374 (2001).

[18] L. Toma and S. Toma, Protein Sci. **5**, 147 (1996).

[19] K. Yue and K.A. Dill, Phys. Rev. E **48**, 2267 (1993); Proc. Natl. Acad. Sci. U.S.A. **92**, 146 (1995).

[20] P. Grassberger, Phys. Rev. E **56**, 3682 (1997).

[21] M.N. Rosenbluth and A.W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).

[22] D. Aldous and U. Vazirani, in *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, edited by (IEEE, New York, 1994), pp. 492–501.

[23] H. Kahn, in *Use of Different Monte Carlo Sampling Techniques*, Symposium on the Monte Carlo Method, edited by H. A. Meyer (Wiley, New York, 1956).

[24] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Series in Statistics (Springer, New York, 2001).

[25] H.-P. Hsu, V. Mehra, W. Nadler, and P. Grassberger (unpublished).

[26] H. Frauenkron, M.S. Causo, and P. Grassberger, Phys. Rev. E **59**, R16 (1999).

[27] In additional runs, we anticipated that contacts between hydrophobic and polar residues should be rare in native states. We thus punished each such contact by multiplying the weight with a factor $q_{HP}<1$. Optimal values of q_{HP} depended on the sequence, but $q_{HP}=0.2$ never was detrimental and typically reduced the CPU times by factors between 2 and 10. For the 103-mer from Refs. [18,29,30], we even found a new lowest energy $E=-55$.

[28] R. König and T. Dandekar, BioSystems **50**, 17 (1999).

[29] K.A. Dill, K. Fiebig, and H.S. Chan, Proc. Natl. Acad. Sci. U.S.A. **90**, 1942 (1993).

[30] E.E. Lattman, K.M. Fiebig, and K.A. Dill, Biochemistry **33**, 6158 (1994).

[31] P. Grassberger, e-print cond-mat/0010265.