Protein Science (1996), 5:2037–2043. Cambridge University Press. Printed in the USA.

A fast conformational search strategy for finding low energy structures of model proteins

THOMAS C. BEUTLER AND KEN A. DILL

Department of Pharmaceutical Chemistry, University of California, San Francisco, California 94143-1204

(RECEIVED April 17, 1996; ACCEPTED July 30, 1996)

Abstract

We describe a new computer algorithm for finding low-energy conformations of proteins. It is a chain-growth method that uses a heuristic bias function to help assemble a hydrophobic core. We call it the Core-directed chain Growth method (CG). We test the CG method on several well-known literature examples of HP lattice model proteins [in which proteins are modeled as sequences of hydrophobic (H) and polar (P) monomers], ranging from 20–64 monomers in two dimensions, and up to 88-mers in three dimensions. Previous nonexhaustive methods—Monte Carlo, a Genetic Algorithm, Hydrophobic Zippers, and Contact Interactions—have been tried on these same model sequences. CG is substantially better at finding the global optima, and avoiding local optima, and it does so in comparable or shorter times. CG finds the global minimum energy of the longest HP lattice model chain for which the global optimum is known, a 3D 88-mer that has only been reachable before by the CHCC complete search method. CG has the potential advantage that it should have nonexponential scaling with chain length. We believe this is a promising method for conformational searching in protein folding algorithms.

Keywords: chain growth algorithm; conformational searching; lattice model; protein folding

The conformational search problem

There have been many important advances on the road to developing a computer protein folding algorithm (Levitt & Warshel, 1975; Kuntz et al., 1976; Wilson & Doniach, 1989; Skolnick & Kolinski, 1990; Covell, 1992, 1994; Sippl et al., 1992; Vajda et al., 1993; Hinds & Levitt, 1994; Kolinski & Skolnick, 1994; Monge et al., 1994; Wallqvist et al., 1994; Boczko & Brooks, 1995; Srinivasan & Rose, 1995; Sun et al., 1995; Yue & Dill, 1996). In order to devise a computer method that can predict the native structure of a protein from its amino acid sequence alone, it is necessary to have an adequate energy function applied to an appropriate chain representation and searched with a fast conformational search method. Currently, the most popular conformational search methods are Molecular Dynamics (MD) and Monte Carlo (MC) and its variants—simulated annealing and genetic algorithms. But these conformational search methods are too slow and "inefficient;" that is, they get stuck in energy traps and are unable to reach the global minima of their energy functions in a reasonable amount of computer time (hours to weeks on workstations). Here we describe a method that improves on the speed and efficiency of existing search methods.

The main problem in developing a conformational search strategy for protein folding is that the energy landscape is large, and sometimes rugged, and we seek the global minima (rather than local minima), of which there are an exceedingly small number. We are searching for a needle in a haystack (Dill, 1993). The success of a search strategy can be judged by two criteria: how deeply it penetrates the energy landscape toward the global minima, and how quickly it gets there.

Exhaustive enumeration methods are guaranteed to reach the global optima, but the computer time τ required to get there increases exponentially with the chain length n, $\tau \approx a^n$, where a is roughly the number of significant rotational isomers per monomer. Smart pruning strategies have been applied in lattice models (Yue & Dill, 1993; Yue et al., 1995) to reduce a from about 5 to 1.125, but the scaling remains exponential, as it does for any strategy guaranteed to find the global optimum.

The most common search methods, MC and MD, are based on sparser sampling of the conformational space, for which they sacrifice the guarantee that they will reach the global minimum. This is not to say that they won't reach the global minima; it is only to say that we have no guarantee whether the minimum found is a global or local one. Indeed some proteins, such as cytochrome *c*

Reprint requests to: Ken A. Dill, Department of Pharmaceutical Chemistry, Box 1204, University of California, San Francisco, California 94143-1204; e-mail: dill@maxwell.ucsf.edu.

(Sosnick et al., 1994) appear to fold along funnel-shaped landscapes. Such proteins are good candidates for such search strategies. However, the time scale for folding cytochrome c is milliseconds, which is 8–12 orders of magnitude longer than the intrinsic time step for conformational change, so even for "funnelscape" protein models, it remains a challenge for a computer to find a global minimum. MC and MD get stuck and become sluggish in the dense compact chain conformations.

There have been many efforts to improve upon MC or MD searches. Unger and Moult (1993) have shown an implementation of a Genetic Algorithm (GA) that is much faster than an implementation of traditional MC in lattice model tests. Methods for fast conformational searching of homopolymers have been developed, including in the dense states. Reptation algorithms and chain-growth algorithms have been used to avoid trapping of the system in local energy minima (Binder & Heermann, 1988). Although reptation is not suitable for heteropolymer problems such as protein folding, chain growth algorithms with "look-ahead" may hold some promise (Rosenbluth & Rosenbluth, 1955; Meirovitch & Livne, 1988). One variant has been adapted by O'Toole and Panagiotopoulos (1992), and our method described below also uses a look-ahead process for the chain growth. However, our approach focuses more on the use of nonlocal, rather than local, energy functions in the heuristics. In the chain-growth algorithms following Rosenbluth and Rosenbluth (1955), the heuristic interaction energies exploit only local information to guide the growth process. External fields have been used to introduce global protein specific information to bias sampling toward low energy structures (Garel & Orland, 1990; Solomon & Liney, 1995).

Two other recently developed conformational sampling methods are Hydrophobic Zippers (HZ) (Fiebig & Dill, 1993) and the Contact Interactions method (CI) (Toma & Toma, 1996). HZ uses a topological definition of "spatial localness" to assemble increasingly nonlocal pairings of monomers to zip up hydrophobic cores. The CI method biases the trial moves in a MC method toward the formation of hydrophobic contacts. CI uses a residue-dependent temperature that depends on a topological distance measure similar to the one used in the HZ algorithm. These methods have the advantage that they have some physical basis; they indicate how β-sheets might form, a difficult problem for traditional search methods; and they have been shown to be much more efficient than traditional MC in several lattice model tests (Yue et al., 1995; Toma & Toma, 1996).

We believe an important principle in enhancing the speed and efficiency of conformational search strategies is to incorporate search biases that are based on some global knowledge of what the potential function is trying to achieve, such as the knowledge that proteins have hydrophobic cores. Some of the methods described above do this to varying degrees. Traditional MC and MD do not use such information. There remains a need for even greater efficiencies in computational search strategies. The present algorithm is intended as a step in this direction. Our approach is based on biasing a chain toward finding a good hydrophobic core. We believe this is the main feature that gives the energy landscape of protein folding much of its overall shape.

The energy function and chain representation

Based on the premise that nonlocal contact interactions dominate folding, the most unambiguous way to test conformational search strategies at the present time is to use the HP lattice model (Lau & Dill, 1989; Chan & Dill, 1991; Dill et al., 1995). Several search strategies have been tested using this model (O'Toole & Panagiotopoulos, 1992; Dill et al., 1993; Fiebig & Dill, 1993; Unger & Moult, 1993; Yue & Dill, 1993; Yue et al., 1995; Hart & Istrail, 1996; Toma & Toma, 1996). In the HP model, proteins are modeled as sequences of hydrophobic (H) and polar (P) monomers. The monomers occupy a string of adjacent sites on a lattice, typically a 2D square lattice or 3D simple cubic lattice. To satisfy excluded volume, each lattice site can be occupied by no more than one monomer. Two H monomers that are adjacent in space, but not adjacent in sequence, are attracted by a contact energy. All other types of interactions are assumed to be zero. Therefore, the globally optimal conformations in this model are simply those with the maximum possible number of HH contacts.

The advantage of using the HP lattice model to test conformational search strategies is that the conformational space is discrete and countable for any sequence, the global minima are unambiguously distinguishable from local minima, the model has been demonstrated to have many protein-like properties (Dill et al., 1995), and the model captures the needle-in-a-haystack nature of the search problem. It is a useful starting standard, because several conformational search methods have already been compared on a small set of well-known test sequences in this model (Unger & Moult, 1993; Yue et al., 1995; Toma & Toma, 1996).

The conformational search strategy

Our Core-directed chain Growth (CG) method grows a chain conformation by a systematic covalent addition of one "segment" at a time. A segment is a connected string of a few residues, the length of which depends on a procedure described below. Our method begins by estimating the size of the hydrophobic core. Following Yue and Dill (1993), we count the total number of H monomers in the sequence, and construct a core, which is as nearly square as possible, that can contain all the H monomers. This is not the final true core of the native protein; it is an optimal core constructed as if there were no chain connectivity constraint, but it gives a framework for construction. We refer to this optimal construct as "the core." The space inside and outside of the core is arranged in shells or layers. Figure 1 illustrates these definitions and shows a growing chain.

The CG algorithm begins by randomly selecting any H monomer, numbered i in the sequence, that has a sequence neighbor $i-1$ or $i+1$ that is also hydrophobic. This first residue is placed randomly on the outermost shell of the core (see Fig. 1).

For the first segment, and all subsequent ones as they are added to the existing chain, we explore all possible segment conformations by exhaustive enumeration, with preceding segment conformations held fixed. Chain growth by segment "look-ahead" has been studied extensively by Meirovitch, particularly for estimating partition functions (Meirovitch, 1977, 1983b, 1983a, 1985; Meirovitch & Livne, 1988). Segment conformations that violate excluded volume constraints (either within the segment or by interference with preexisting segments) are eliminated. Among the remaining acceptable segment conformations, a conformation of the newly added segment is chosen according to a weight computed from the following heuristic function, f_h:

$$f_h = \sum_{H \in segment} S_H^{field} + \sum_{P \in segment} S_H^{field} + \sum_{HH\ contacts} S_{HH} + \sum_{PP\ contacts} S_{PP} + \sum_{HP\ contacts} S_{HP}. \quad (1)$$

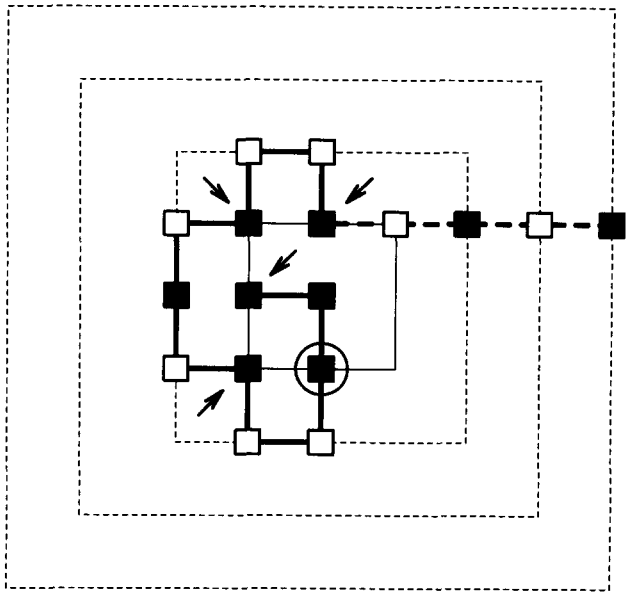

Fig. 1. A growing chain shown superimposed on its lattice core and surrounding layers. Thin full line indicates the core. Thin dashed lines indicate the shells. Chain segments already laid down and fixed into place are indicated with heavy line bonds. Dashed bonds indicate a chain segment for which an exhaustive search is being performed during the current growth step. Circled residue is the first residue that was laid on the lattice. Arrows point to end residues of previously grown chain segments. Minimum segment length was chosen to be three residues in this example.

The main task of the heuristic function is to foster the formation of a hydrophobic core. In f_h, the first two terms are "unitary" based on propensities of H or P to be in a core, and the last terms are "binary" based on pair interactions: HH, HP, or PP. The unitary terms describe a sort of "mean-field" bias to move H monomers toward the core and to move P monomers toward the surface. Table 1 shows the components of the scoring function (positive means favorable). This function does not preclude P's inside or H's outside. The scoring function does not influence the *energy* of a conformation; the bias function simply chooses which conformations to explore first.

The binary parameters favor HH contacts, in a "time-dependent" or "growth-dependent" way. At first, the HH score, S_{HH}, is set to zero, to prevent premature hydrophobic collapse. At later stages, the HH attraction is strengthened. Because any PH contacts prohibit potential HH contacts, we set $S_{PH} < 0$ from the beginning.

The function f_h is highly degenerate: many different conformations have the same score f_h. We choose among the highest-scoring conformations randomly.

After fixing the conformation of a segment, we choose the next segment by stepping along the sequence toward one of the two chain ends. We choose randomly between the two possible growth directions until a chain terminus is reached; then growth is unidirectional. For each added segment, we must choose its length. The segment length, $l_{segment}$, is chosen to be between an upper limit, $l^{max}_{segment}$, and a lower limit, $l^{min}_{segment}$. A segment is chosen to be long enough to reach the next H in the sequence, or to have the upper limit length, whichever is shorter. The aim is to have H as the last residue in the segment because the core region is smaller than the non-core region and contacts with H monomers are stronger determinants of structure than those with P monomers. The segment length minimum is needed to insure that conformational searching is not too local. This cycle of choosing a growth direction, then a segment length, then enumerating all segment conformations and selecting a good one, is repeated until the chain is fully grown (see Fig. 2).

After laying down a full-chain conformation, we use a second phase of iterative refinement for those chains that reached a sufficiently large number of HH contacts. We chose the best 1% of the conformations for refinement. For the chosen conformations, we delete backward from the end until we reach a randomly chosen nucleus of length l_{nuc}. Then the algorithm regrows the rest of the chain in the same manner as described above. This second phase gives a substantial improvement in the search power. We stop the algorithm after some tolerable period of computer time. There is

Table 1. *Parameters of the heuristic function*

Parameter	Value	Description
S_H^{field}	1	Weight of H in core region.
	0, −1, . . .[a]	Weight of H outside core region.
S_P^{field}	1	Weight of P outside core region.
	0	Weight of P in first shell inside core region.
	−1	Weight of P in the deeper shells of the core region.
S_{HH}	0	Weight of HH contacts while $l < 1/3\ l_{chain}$.[b]
	1	Weight of HH contacts while $1/3\ l_{chain} \leq l < 3/4\ l_{chain}$.
	2	Weight of HH contacts while $l \geq 3/4\ l_{chain}$.
S_{PP}	0	Weight of PP contacts.
S_{HP}	−1	Weight of HP contacts.
$l^{min}_{segment}$	2	Minimum segment length in growth step.
$l^{max}_{segment}$	5	Maximum segment length in growth step.
l_{nuc}	$1/3\ l_{chain}$	Length of the segments left as nuclei in the second phase.

[a] Weight is decremented by −1 in each lattice shell surrounding the estimated core region.
[b] l, Current length of chain. l_{chain}, Length of complete chain.

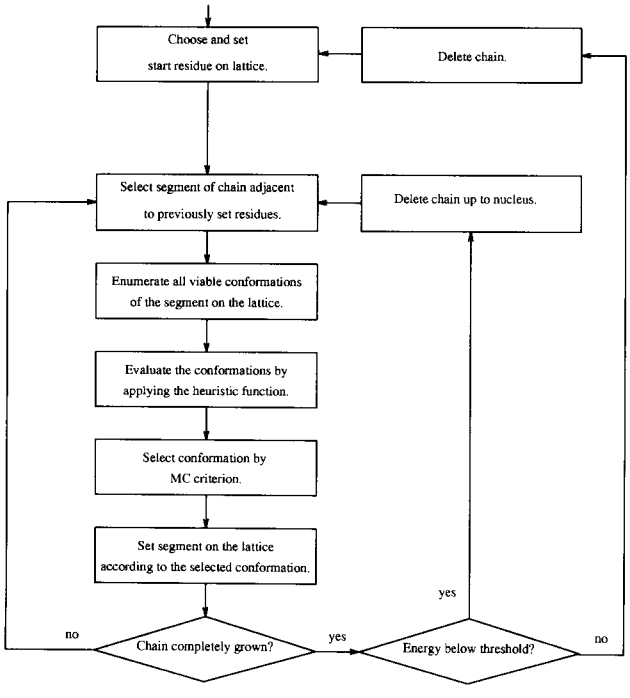

Fig. 2. CG algorithm.

no guarantee that CG will find global minima. But we find (see below) that CG finds global optima more reliably than other methods that have been tested on the same HP model sequences.

Tests of search speed and penetration depth

Table 2 shows 2D lattice model tests on sequences generated by Unger and Moult (1993). Using those test sequences, Unger and Moult showed that their GA was much faster than their MC search strategy. Using the same test sequences, Toma and Toma (1996) recently showed that the CI method was faster than the GA method of Unger and Moult. Also, the CI method of Toma and Toma found the 48-mer and 60-mer structures that the GA had failed to find previously. Table 2 compares our CG results with the GA and CI results. The times given for the GA and the CI method are estimates of the CPU time needed by a Sparc 1 workstation based on the reported number of energy evaluations. The times given for CG correspond to the measured CPU times on a Sparc 1.

Table 2. *Performance of the CG method in comparison to results reported in the literature*

l[a]	E_{min}[b]	t_{GA} (min)[c]	t_{CI} (min)[d]	t_{CG} (min)[e]
20	−9	8.6×10^{-2}	4.8×10^{-4}	3.6×10^{-2}
24	−9	1.0×10^{-1}	4.0×10^{-3}	1.1×10^{-1}
30	−8	6.1×10^{-2}	3.0×10^{-3}	7.8×10^{-2}
36	−14	9.1×10^{-1}	1.1×10^{-1}	2.2×10^{0}
48	−23	—	5.8×10^{-1}	6.3×10^{0}
50	−21	5.3×10^{1}	0.4×10^{-2}	3.1×10^{2}
60	−35	—	1.0×10^{0}	9.7×10^{1}
64	−42	—	—	9.1×10^{0}

[a]Sequence length.
[b]Global energy minimum in units of number of HH contacts.
[c]Estimate of CPU time required to find the minima using a GA for the sequences for which the minima were found (Unger & Moult, 1993).
[d]Estimate of CPU time required to find the minima using the CI Algorithm for the sequences where the method succeeded (Toma & Toma, 1996).
[e]Average CPU time needed to find the global energy minimum using CG.

Table 2 shows that the CG search strategy penetrates to lower free energies than the GA or CI methods. The CG approach finds the conformations of minimum free energy in all cases. In contrast, GA does not find the 48-mer, 60-mer, or 64-mer structures, and the CI method does not find the 64-mer. Moreover, the CG method discovered that the 48-mer and 60-mer structures that Unger and Moult believed were the global minima are, in fact, only local minima. Our CG search finds structures of lower energy in those two cases. For these short sequences in 2D, our method is slower than CI, and roughly comparable to GA (although our comparisons are somewhat different: Unger & Moult [1993] and Toma & Toma [1996] report the best speed in five runs, whereas here we report average speeds). If we define the mean "search velocity" as the total depth of the free energy landscape (to the global minimum) divided by the time required to get there, it is clear that all these methods give very high search velocities for the short chains (9 contacts/3.6×10^{-2} = 250 HH contacts per minute for the 20-mer, compared to 4.6 contacts per minute for the 64-mer). As noted below, the CG method increasingly overtakes the GA and CI methods for longer chain lengths, particularly in three dimensions.

Figure 3 shows the chain-length dependence of the average CPU time required by CG to find the global minimum on a Sparc 1 workstation. This figure shows that the dependence of the search time on sequence is often greater than the dependence on chain length. In this case, the longer sequences in the test set of Unger and Moult need less search time by CG than would be predicted from the shorter sequences.

Table 3 shows longer chain 3D test cases. Table 3 gives the search times for the 10 different 3D 48-mer sequences used by Yue et al. (1995) to compare MC and HZ search methods. We show also the computer times for the constraint-based exhaustive method

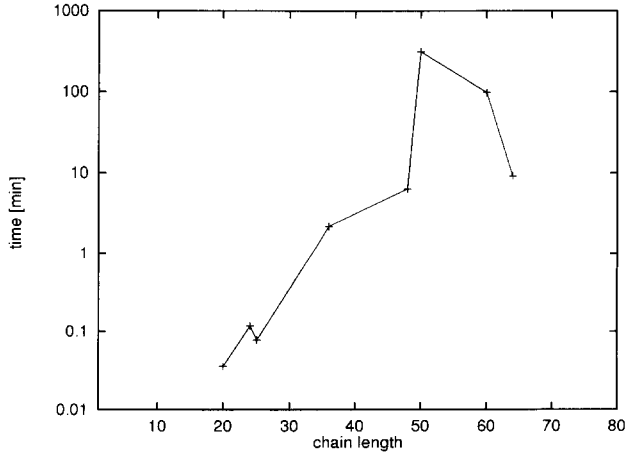

Fig. 3. CG search speed versus chain length. Average CPU time (Sparc 1) needed to find the global energy minimum, for the 2D HP lattice model proteins of Unger and Moult (1993).

Table 3. *Performance of the CG method in comparison to MC, HZ, and CHCC in finding the global energy minima of HP model 48-mer proteins on 3D cubic lattices*

No.[a]	E_{MC}[b]	E_{HZ}[c]	E_{min}[d]	t_{HZ} (min)[e]	t_{CHCC} (min)[f]	t_{CG} (min)[g]
1	−30	−31	−32	1.5×10^4	3.0×10^1	9.4×10^0
2	−30	−32	−34	7.1×10^4	2.3×10^0	3.5×10^1
3	−31	−31	−34	8.2×10^4	3.0×10^1	6.2×10^1
4	−30	−30	−33	1.6×10^6	7.1×10^1	2.9×10^1
5	−30	−30	−32	1.1×10^5	3.2×10^1	1.2×10^1
6	−30	−29	−32	1.8×10^5	8.0×10^1	4.6×10^2
7	−31	−29	−32	2.2×10^4	1.1×10^1	6.4×10^1
8	−31	−29	−31	1.4×10^4	5.3×10^1	3.8×10^1
9	−30	−31	−34	1.6×10^4	8.3×10^0	—[h]
10	−30	−33	−33	4.1×10^3	4.8×10^0	1.1×10^0

[a]Sequence number corresponds to sequences and numbers given in Yue et al. (1995).
[b]Minimum energy obtained by MC, as given in Yue et al. (1995) (one HH contact corresponds to one energy unit).
[c]Minimum energy obtained by applying HZ algorithm, as given in Yue et al. (1995).
[d]Global energy minimum as determined by Yue et al. (1995).
[e]Estimate of CPU time needed to find the global energy minimum using HZ algorithm.
[f]CPU time needed by CHCC (Yue et al., 1995).
[g]Average CPU time needed to find the global energy minimum using CG.
[h]Global minimum was not reached within 72 h, but $E = 33$ is reached in 26 min.

(CHCC, Yue & Dill, 1993; Yue et al., 1995). CHCC is currently the fastest lattice method that is guaranteed to find the global minimum. The two left-hand energy columns show that the MC and HZ methods each found a global minimum for only 1 of the 10 sequences. Toma and Toma (1996) reported that the CI method finds global minima in 2 of the 10 cases. Our CG method finds global minima for 9 of the 10 sequences. For sequence 9, where CG fails to find the global minimum in 72 h, it finds a first excited state in 26 min. Therefore, although it is not perfect, the CG method is the only nonexhaustive method currently available that is reasonably reliable in finding global minima for these 3D 48-mers. It is worth noting that reaching global optima, by any method, normally takes far longer than reaching near-global optima because energy landscapes become very narrow as a function of depth. In that regard, the most challenging test of a search method is whether it finds global optima, rather than just local optima. CG performs very well in this regard.

Moreover, the search times for CG are about as short as those of CHCC, and a few orders of magnitude faster than our estimate of how long HZ would have taken to find the true minima. It is

No	*Sequences*[a]	
1	HPHPPHHPHPPHPHHPPHPH	
2	HHPPHPPHPPHPPHPPHPPHPPHH	
3	PPHPPHHPPPPHHPPPPHHPPPPHH	
4	PPPHHPPHHPPPPPHHHHHHHHPPHHPPPPHHPPHPP	
5	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH	
6	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH	
7	PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHHHHHHHHHHHPPPPHHHH HHPHHPHP	
8	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHPPHHPPHPPHHPPHPHPHHHHHH HHHHHH	
No	*Examples of globally optimal conformations*[b]	E^c_{min}
1	RFFLBLFLLBRBRBLBRRF	9
2	BLBRBRFRFLFFFRFLFLBLBRB	9
3	FRBRFRRRFLLFRFLLBLLLBRRR	8
4	LLLFRFLFFFLLBBRBLBRBBLFLLLFRRFLFRFF	14
5	BBRBLBRBLBBBLLFFRFLFFLBBBBBBLFFFFLFRFLFRFRBRRB	23
6	RBLLFFRFRBRFFLFRRRFRRRFFRBBRBLLBLLLLBBRFRBBLBLL FR	21
7	BBLLLBRRRBLLLLBBRFRRRBLLBLBRBRFFRRFFFFRBBBBRRFFLFFFLLFRFR BR	35
8	FLFRRBBRFRRBLBLBLFLBLFLBLFLFRFLFRFLFRFRBRFRBRFRBRBRBLLFLLLLBBBR	42

Fig. 4. HP test sequences and their native conformations in two dimensions. [a]Sequences taken from Unger and Moult (1993). [b]L, left; R, right; F, forward; B, backward. [c]Energy of globally optimal conformations.

No	*Sequences*[a]
1	HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
2	HHHHPHHPHHHHHPPHPPHHPPHPPPPPPPHPPHPPPHPPHHPPHHHPH
3	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP
4	PHPHHPPHPHHHPPHHPHHPPPHHHHHHPPHPHHPHPHPPPPHPPHPHP
5	PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPP PPHHPHHPH
6	HHHPPPHHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHP PHHHHHHPH
7	PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPH HPPHHPPPH
8	PHHPHHHPHHHHPPHHHPPPPPPHPHHPPHHPHPPPHHP HPHPHHPPP
9	PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPH PHHHPPPPH
10	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHH PPHPHPPHP

No	*Examples of globally optimal conformations*[b]	E^c_{min}
1	FLLDBURBURFLFFRBUUBDBLFFULBDFFDBBLBURBBLLDRRFFD	32
2	FLBLFDBRRFLFFLBLURRFRBRRFDBLLFDBDLLUFRBBDRUBL LF	34
3	FDFLBUFRFRBRBLDFDFUFULDBLULUBRURDRUFDLULDFRUR DF	34
4	RDRFFUBLURRFDBDRUUUBLDDLUBURDRFDBLDFDLBUUBLF UFU	33
5	DLUBLDLFRFLFDBRBBLFDBRFFDLUFRRBDBURRFUBLLFFLU RB	32
6	FLLFURRUBLDRRDBULLURRRRDDDLLFLDBUBLUFLDRFFURD RU	32
7	RDBBRULFRFFURBLBLURFLDFFURBRBRDDLBURULBBDFDB LUF	32
8	BBBRUFFDBDFFRBBURRFFLLBUBBBDFDDFDFLUBBULFDFUF FD	31
9	ULULLURRFDDRFUBUBDRFUBUBDDLBRBUFLFLLDRDRRFDR FUL	34[d]
10	URFUBLFFRUBULDLDFURFRDDBLBDRRUBUFUUFDDRDLFUFUBU	33

Fig. 5. 48-mers in three dimensions. [a]Sequences taken from Yue et al. (1995). [b]U: up, D: down, L: left, R: right. [c]Energy of globally optimal conformations. [d]Minimum energy not reached by CG.

known that each additional HH contact, close to the global minimum, requires an exponentially increasing search time, so the column t_{HZ} in Table 3 is estimated on that basis. CG has the disadvantage compared to CHCC that it does not guarantee the global optimum, but the possible advantage that, because it is not exhaustive and therefore should not scale exponentially with chain length, it may be useful for much longer chains. The problem is that there are very few longer-chain test cases currently available. The longest chain test case available for which a provable global minimum is known is an 88-mer searched by the CHCC method (K. Yue, pers. comm.). CHCC finds global optima for the 88-mer in about 40 min and the CG method finds a global minimum for that sequence within 72 h on a Sparc 10 workstation.

Figures 4, 5, and 6 list the test sequences we used, along with some examples of globally optimal conformations.

The superior efficiency of the CG algorithm compared to the HZ algorithm is also shown in Figure 7, where the lowest energies found in 10-h CPU time on a Sparc 1 workstation are shown. The CG method uniformly reaches conformations with substantially lower energies than HZ in that time, and the relative performance increases with chain length.

The choice of the parameter values is not particularly critical. The values in Table 1 were used in all the cases we reported here. We have not attempted to optimize them. It may be possible to improve the search efficiencies.

No	*Sequence*[a]
1	PHPHHPHHPHPPHHPPHPPHPPHPPHPPHPPHHPPHHHPPHHHPP HH HPPHHHPPHPHHPHHPHPPHPPHPPHHPPHPPHPPHHPPH P

No	*Example of a globally optimal conformation*[b]	E^c_{min}
1	URFUBLURFRFLDRDLDFURULFUBULDFDBDDBUUUCBDDDDLUUU UFDDDDFULBUFRULBLBRUBDBRFDBDFDLULFRULBRB	72

Fig. 6. 88-mer in three dimensions. [a]K. Yue (pers. comm.). [b]U, up; D, down; L, left; R, right; F, forward; B, backward. [c]Energy of globally optimal conformation.

Conclusions

We have described a new conformational search method that we call Core-directed chain Growth (CG). For several 2D and 3D lattice model literature test sequences, we find that CG substantially outperforms other nonexhaustive methods, according to published literature results, including various implementations of MC, GA, CI, and HZ. CG is effective in finding conformations of very low free energy, finding global optima in 90% of the published 3D 48-mer structures, in contrast to all other nonexhaustive methods, which find only 20% or less. The algorithm owes its efficiency to the use of biasing interactions that guide the chain growth to favor hydrophobic burial while they prevent clustering of neighboring hydrophobic residues early in the growth process, because these often impede nonlocal contacts characteristic of very low-energy structures.

The CG method is even competitive in efficiency with CHCC, the complete search algorithm of Yue and Dill (1993). Possible advantages are that: (1) the search time in the CG method may not scale exponentially with the chain length, as CHCC does, and thus might reach longer chain lengths in reasonable computer time; (2) CG does not rely on HP lattice-specific features, and therefore should generalize readily to more realistic protein folding models; and (3) these ideas are simple to implement.

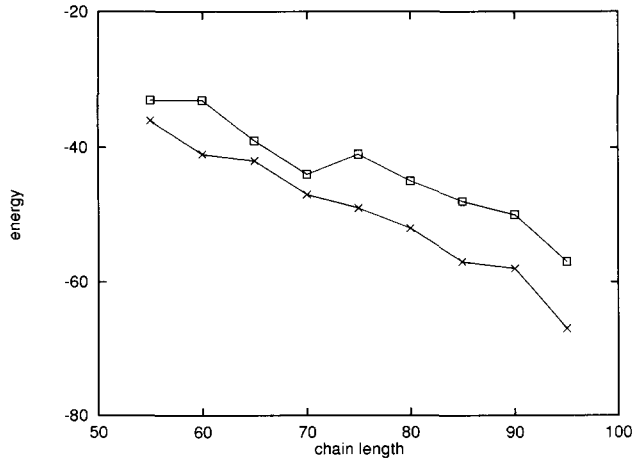

Fig. 7. Search depth of CG (crosses) versus HZ (boxes) for 3D HP lattice model proteins in 10 h CPU time on a Sparc 1 workstation. One HH contact corresponds to one energy unit. CG reaches lower free energies in a given search time.

Acknowledgments

We thank Kai Yue for giving us the native conformations of the 88-mer sequence and for providing information on the efficiency of his algorithm. We thank Klaus Fiebig for providing the HZ program used to generate the data in Figure 7. Financial support was obtained from the Schweizerischen Nationalfonds, which is gratefully acknowledged.

References

Binder K, Heermann DW. 1988. *Monte Carlo simulations in statistical physics.* Berlin/New York: Springer.

Boczko EM, Brooks CL. 1995. First-principles calculation of the folding free energy of a three-helix bundle protein. *Science 269*:393–396.

Chan HS, Dill KA. 1991. "Sequence space soup" of proteins and copolymers. *J Chem Phys 95*:3775–3787.

Covell DG. 1992. Folding protein α-carbon chains into compact forms by Monte Carlo methods. *Proteins Struct Funct Genet 14*:409–420.

Covell DG. 1994. Lattice model simulations of polypeptide chain folding. *J Mol Biol 235*:1032–1043.

Dill KA. 1993. Folding proteins: Finding a needle in a haystack. *Curr Opin Struct Biol 3*:99–103.

Dill KA, Bromberg S, Yue K, Fiebig KM, Yee DP, Thomas PD, Chan HS. 1995. Principles of protein folding—A perspective from simple exact models. *Protein Sci 4*:561–602.

Dill KA, Fiebig KM, Chan HS. 1993. Cooperativity in protein folding kinetics. *Proc Natl Acad Sci USA 90*:748–752.

Fiebig KM, Dill KA. 1993. Protein core assembly process. *J Chem Phys 98*:3475–3487.

Garel T, Orland H. 1990. Guided replication of random chains: A new Monte Carlo method. *J Phys A 23*:L621–L626.

Hart WE, Istrail S. 1996. Fast protein folding in the hydrophobic–hydrophilic model within three-eights of optimal. *J Comput Biol.* Forthcoming.

Hinds D, Levitt M. 1994. Exploring conformational space with a simple lattice model for protein structure. *J Mol Biol 243*:668–682.

Kolinski A, Skolnick J. 1994. Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme. *Proteins Struct Funct Genet 18*:338–352.

Kuntz ID, Crippen GM, Kollman PA, Kimelman D. 1976. Calculation of protein tertiary structure. *J Mol Biol 106*:983–994.

Lau KF, Dill KA. 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules 22*:3986–3997.

Levitt M, Warshel A. 1975. Computer simulation of protein folding. *Nature 253*:694–698.

Meirovitch H. 1977. Calculation of entropy with computer simulation methods. *Chem Phys Lett 45*:389–392.

Meirovitch H. 1983a. Improved computer simulation method for estimating the entropy of macromolecules with hard-core potential. *Macromolecules 16*:1628–1631.

Meirovitch H. 1983b. Method for estimating the entropy of macromolecules with computer simulation. Chains with excluded volume. *Macromolecules 16*:249–252.

Meirovitch H. 1985. Scanning method as an unbiased simulation technique and its application to the study of self-attracting random walks. *Phys Rev A 32*:3699–3708.

Meirovitch H, Livne S. 1988. Computer simulation of long polymers adsorbed on a surface. I. Corrections to scaling in an ideal chain. *J Chem Phys 88*:4498–4506.

Monge A, Friesner RA, Honig B. 1994. An algorithm to generate low-resolution protein tertiary structures from knowledge of secondary structure. *Proc Natl Acad Sci USA 91*:5027–5029.

O'Toole EM, Panagiotopoulos AZ. 1992. Monte Carlo simulation of folding transitions of simple model proteins using a chain growth algorithm. *J Chem Phys 97*:8644–8652.

Rosenbluth MN, Rosenbluth AW. 1955. Monte Carlo calculation of the average extension of molecular chains. *J Chem Phys 23*:356–359.

Sippl M, Hendlich M, Lackner P. 1992. Assembly of polypeptide and protein backbone conformations from low energy ensembles of short fragments: Development of strategies and construction of models for myoglobin, lysozyme, and thymosin beta 4. *Protein Sci 1*:625–640.

Skolnick J, Kolinski A. 1990. Simulations of the folding of a globular protein. *Science 250*:1121–1125.

Solomon JE, Liney D. 1995. Exploration of compact protein conformations using the guided replication Monte Carlo method. *Biopolymers 36*:579–597.

Sosnick TR, Mayne L, Hiller R, Englander SW. 1994. The barriers in protein folding. *Nature Struct Biol 1*:149–156.

Srinivasan R, Rose GD. 1995. LINUS—A hierachic procedure to predict the fold of a protein. *Proteins Struct Funct Genet 22*:81–99.

Sun S, Thomas PD, Dill KA. 1995. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Eng 8*:769–778.

Toma L, Toma S. 1996. Contact interactions method: A new algorithm for protein folding simulations. *Protein Sci 5*:147–153.

Unger R, Moult J. 1993. Genetic algorithms for protein folding simulations. *J Mol Biol 231*:75–81.

Vajda S, Jafri MS, Sezerman OU, DeLisi C. 1993. Necessary conditions for avoiding incorrect polypeptide folds in conformational search by energy minimization. *Biopolymers 33*:173–192.

Wallqvist A, Ullner M, Covell DG. 1994. A simplified amino acid potential for use in structure predictions of proteins. *Proteins Struct Funct Genet 18*:267–280.

Wilson C, Doniach S. 1989. A computer model to dynamically simulate protein folding—Studies with crambin. *Proteins Struct Funct Genet 6*:193–209.

Yue K, Dill KA. 1993. Sequence–structure relationships in proteins and copolymers. *Phys Rev E 48*(3):2267–2278.

Yue K, Dill KA. 1996. Folding proteins with a simple energy function and extensive conformational searching. *Protein Sci 5*:254–261.

Yue K, Fiebig KM, Thomas PD, Chan HS, Shakhnovich EI, Dill KA. 1995. A test of lattice protein folding algorithms. *Proc Natl Acad Sci USA 92*:325–329.