J. Mol. Biol. (1993) **231**, 75–81

Genetic Algorithms for Protein Folding Simulations

Ron Unger[1,2] and John Moult[1]

[1]*Center for Advanced Research in Biotechnology*
Maryland Biotechnology Institute, University of Maryland
9600 Gudelsky Drive, Rockville, MD 20850, U.S.A.

[2]*Institute for Advanced Computer Studies*
University of Maryland, College Park, MD 20742, U.S.A.

(Received 14 July 1992; accepted 18 September 1992)

Genetic algorithms methods utilize the same optimization procedures as natural genetic evolution, in which a population is gradually improved by selection. We have developed a genetic algorithm search procedure suitable for use in protein folding simulations. A population of conformations of the polypeptide chain is maintained, and conformations are changed by mutation, in the form of conventional Monte Carlo steps, and crossovers in which parts of the polypeptide chain are interchanged between conformations. For folding on a simple two-dimensional lattice it is found that the genetic algorithm is dramatically superior to conventional Monte Carlo methods.

Keywords: protein folding simulations; genetic algorithms; lattice models; search methods; folding pathways

1. Introduction

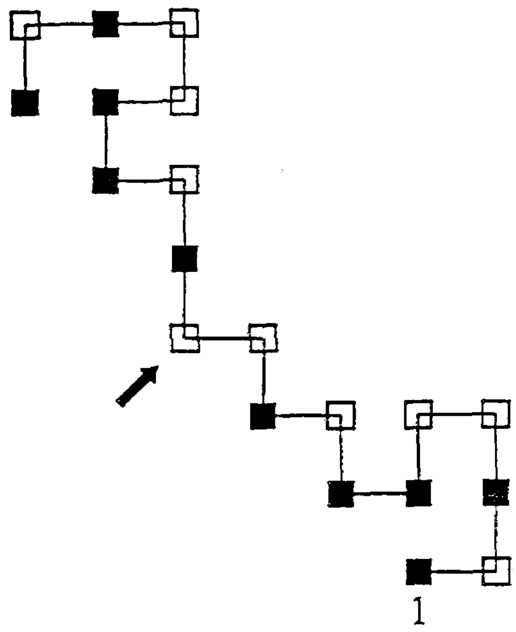
Computing the functional conformation of a protein molecule from the amino acid sequence is difficult for two reasons: the contributions to free energy

mutations and crossovers. The latter process is the heart of the method. Technically, the operation consists of exchanging parts of strings between pairs of solutions, so as to yield new solutions. This has a the usual manner, in a process equivalent to the accumulation of point mutations. Then selected polypeptide chains are cut and each rejoined to another chain cut at the same point (crossovers). Metropolis-type criteria are used to see if each newly generated conformation should be accepted. Those that are accepted enter the MC phase again, and the process is iterated. Here, we describe the details of the procedure and compare its effectiveness with Monte Carlo alone. We find that a simple GA can dramatically improve search effectiveness in a model of protein folding.

2. The Model

We wish to develop an implementation of a GA suitable for protein folding and compare it with the MC method. Thus, we seek to use the simplest model that still captures the essence of the important components of protein folding (Lau & Dill, 1990). The linear sequence is composed of "amino-acids" of only two types: hydrophobic (black) and hydrophilic (white). This sequence is "folded" on a two-dimensional square lattice on which at each point the chain can turn 90° left or right, or continue ahead. The energy function is simple: -1 for each direct contact (occupying neighboring non-diagonal lattice points) of non-bonded hydrophobic-hydrophobic amino acids. Figure 1 shows possible conformations of the 20 amino acid molecule *B-W-B-W-W-B-B-W-B-W-W-B-W-B-B-W-W-B-W-B*.

Under this energy function, low energy con-



(A)

-4

Table 1
Energy level distribution

Energy level	No. of conformations
0	36,098,079
−1	31,656,934
−2	12,473,446
−3	2,943,974
−4	517,984
−5	77,080
−6	10,364
−7	1194
−8	96
−9	4
Total	83,779,155

A full enumeration was performed to evaluate the energy of all ... self-avoiding confirmations possible for the sequence BWBW-BBWBWWBWBBWWBWB. For each energy level we list the number of conformations with that energy. Note that the largest fractional decrease is between the number of conformations found in energy level −8 and the number of conformations with the lowest energy level −9.

possibilities is exponential in the length of the sequence. Our goal is to devise a search algorithm that can find a conformation with the lowest free energy value. For the sequence given above, the energies of all the 83,779,155 possible valid conformations were calculated (see Table 1). The number of conformations in each energy level decreases rapidly, with the largest fractional decrease in the final transition to the lowest energy level: there are four conformations with energy −9 *versus* 96 conformations with −8. (Similar behavior was observed for 24 residue long sequences.) Note that even for this very simple lattice model the precise arrangement of an optimal conformation is very rare and difficult to achieve. The infinitesimally small size of the optimal subset relative to the size of the conformational space (*only* $\simeq 0{\cdot}5 \times 10^{-7}$ of the conformations!) highlights the problem of designing an efficient search.

accepted, then retain the former conformation S_1. (4) If the stop criterion is not met, then repeat steps (2) to (4).

Theoretically, with the appropriate cooling scheme this algorithm is guaranteed to converge to the global minimum, but it must be remembered that the number of steps in such an "appropriate" scheme is strikingly large. It is actually larger than the exponential number of steps needed to enumerate the whole space! (The theoretical aspects of MC methods are discussed in Aarts & Korst (1989), chapter 3.) Practically, the selection of the cooling scheme is crucial for the success of the process. Usually, c_k is cooled linearly (i.e. $c_{k+1} = \alpha c_k$, where α is a constant smaller than but close to 1). As the minimum energy value is not known in advance and as the algorithm does not always converge to the lowest energy level it has encountered, the usual procedure is to run the algorithm as long as the computer resources permit, while decreasing c_k gradually and keeping track of the lowest energy solution found.

In our model the initial conformation is fully extended (i.e. a straight line). The random change is performed by randomly selecting an amino acid and rotating the C-terminal portion of the chain around that amino acid (see Fig. 1). For the 20 amino acid example above, the algorithm was run for 50,000,000 steps, about one half of which yielded valid (self-avoiding) conformations. When a valid conformation was encountered its energy was evaluated. The c_k was reduced very slowly from 2 to 0·15 (c_k was decreased by $\alpha = 0{\cdot}99$ every 200,000 steps), reducing the chance of accepting a move with a cost of +1 from 0·6 to 10^{-3}. The simulation was run five times. In these runs, an optimal conformation with energy −9 was found after 3,199,813, 8,823,199, 469,984, 292,443 and 7,367,375 energy evaluations, respectively.

4. Genetic Algorithms

In implementing a genetic algorithm, one has to choose the appropriate method of encoding the

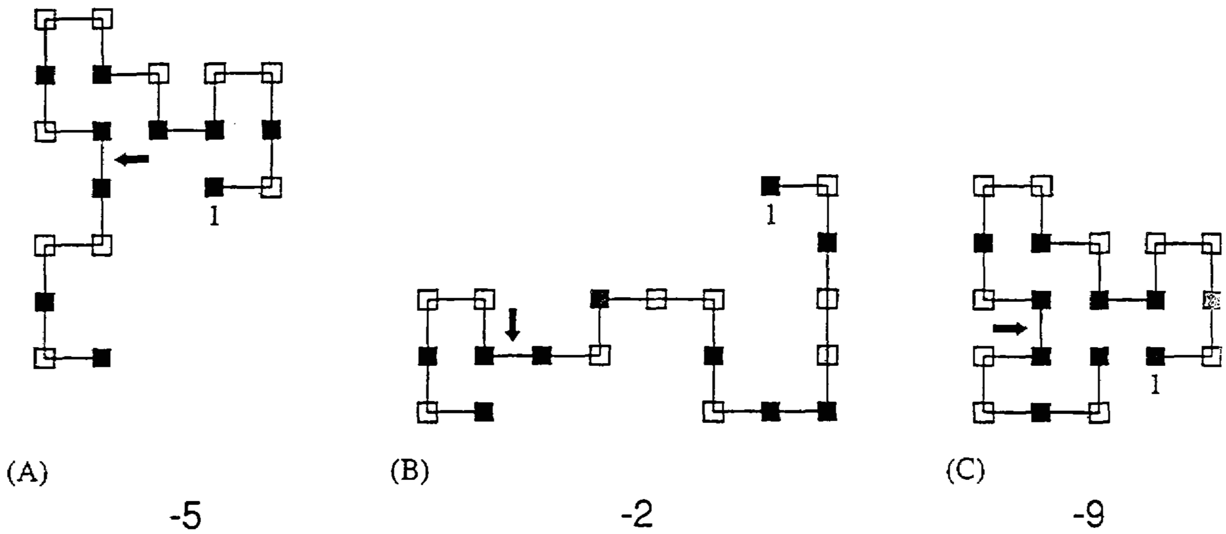


Figure 2. The genetic algorithm. The process starts with a population of fully extended structures. Each structure undergoes a MC stage followed by a crossover stage. In the crossover stage, pairs of structures are randomly (based on their energies) cut and pasted. In this example the cutpoint was randomly chosen to be after residue 14. Joining the first 14 residues of (A) with the last 6 residues of (B) and applying a randomly chosen 270° rotation at the joint achieves the compact structure in (C). In this case, the energy value of the hybrid (C) is -9, lower than the energies -5 and -2 of its "parents". The hybrid is always accepted if its energy is lower than the averaged energies of its parents, or non-deterministically accepted according to its energy increase.

Thus, the lower energy conformations have a higher chance of being selected. For a pair of selected structures a random point is chosen along the sequence and the N-terminal portion of the first structure is connected to the C-terminal portion of the second structure (see Fig. 2). As there are three ways to join the parts together (connecting the chains with angles of 0°, 90° or 270°), these possibilities are tested in a random order to find one that

stages. Five of the structures after the fifth and the tenth generations are shown in Figure 3. Each application of a genetic operator is counted as a step. Thus, a generation takes $20 \times 200 = 4000$ mutation steps plus the number of crossover trials it takes to get 200 new valid structures, typically around 900 steps. When a valid conformation is encountered, its energy is evaluated. The simulation was run for five times. The optimal conformation

Table 2
Comparison of genetic algorithm (GA) and Monte Carlo (MC) folding simulations

Length[a]	Optimal energy[b]	GA[c]	MC[d]	Long MC[e]	Multiple MC[f]
20	−9	−9 (30,492)	−8	−9 (292,443)	−9 (41%)
24	−9	−9 (30,491)	−8	−9 (2,492,221)	−9 (19%)
25	−8	−8 (20,400)	−7	−8 (2,694,572)	−7 (100%)
36	−14	−14 (301,339)	−12	−13 (6,557,189)	−13 (5%)
48	−22	−22 (126,547)	−18	−20 (9,201,755)	−19 (3%)
50	−21	−21 (592,887)	−19	−21 (15,151,203)	−20 (1%)
60	−34	−34 (208,781)	−31	−33 (8,262,338)	−32 (7%)
64	−42	−37 (187,393)	−31	−35 (7,848,952)	−32 (2%)

Eight sequences were tested. For each, the GA method is compared with several MC variants. We check a MC that uses as many energy evaluations as the GA, a much longer MC and a multiple MC running 100 different simulations. The GA results are very superior to those from the comparable MC which failed to find the optimal solution in all cases, and are also better than the longer MC runs. In the 36, 48 and 60-residue long sequences, the lowest energy conformation was found by the GA and was not found in any of the MCs. All the methods failed to find the lowest energy conformation for the longest sequence.

[a] The following sequences were tested:

(20) BWBWWBBWBWWBWBBWWBWB;
(24) BBWWBWWBWWBWWBWWBWWBWWBB;
(25) WWBWWBBWWWWBBWWWWBBWWWWBB;
(36) WWWBBWWBBWWWWWBBBBBBBWWBBWWWWBBWWBWW;
(48) WWBWWBBWWBBWWWWWBBBBBBBBBBWWWWWWBBWWBBWWBWWBBBBB;
(50) BBWBWBWBWBBBBWBWWWBWWWBWWWWBWWWBWWWBWBBBBWBWBWBWBB;
(60) WWBBBWBBBBBBBBWWWBBBBBBBBBBWBWWWBBBBBBBBBBBBWWWWBBBBBBWBBWBW;
(64) BBBBBBBBBBBBWBWBWWBBWWBBWWBWWBBWWBBWWBWWBBWWBBWWBWBWBBBBBBBBBBBB.

[b] The optimal energies were determined from the designed structures. For the first 2 sequences these energies were validated by full enumeration of the energies of all valid conformations.

[c] The GA was run with 200 structures for 300 generations. For the mutation stage the cooling scheme starts with $c_k = 2$ and is cooled by $c_k = 0{\cdot}97c_k$ every 5 generations. The crossover stage starts with $c_k = 0{\cdot}3$ and is cooled by $c_k = 0{\cdot}99c_k$ every 5 generations. For each sequence the simulation was run 5 times. For the most efficient run we report the lowest energy value achieved together with the number of conformations scanned before that value was found.

[d] The MC was run to scan the number of conformations given in the GA column. c_k starts as 2 and decreases as $c_k = 0{\cdot}95c_k$ every 1/50 of the number of conformations. The simulation was repeated 5 times, and the lowest energy value found during these simulations is listed.

[e] [illegible] MC [illegible] 50,000,000 steps [illegible] 200,000 steps). The simulation was run 5 times. The lowest [illegible]

amino acid sequences over discrete conformation spaces. *Biochemistry*, **30**, 4232–4237.

Davidor, T. (1990). *Genetic Algorithms and Robotics: A Heuristic Strategy of Optimization*, World Scientific, New Jersey.

Dill, K. A. (1990). Dominant forces in protein folding.

Lau, K. F. & Dill, K. A. (1990). Theory for protein mutability and biogenesis. *Proc. Nat. Acad. Sci., U.S.A.* **87**, 638–642.

Levinthal, C. (1968). Are there pathways for protein folding? *J. Chem. Phys.* **65**, 44–45.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N.,