BULLETIN

OF THE

KOREAN CHEMICAL SOCIETY

ISSN 0253-2964
Volume 21, Number 3

BKCSDE 21(3) 277-364
March 20, 2000

Articles

Comparison of Recombination Methods and Cooling Factors in Genetic Algorithms Applied to Folding of Protein Model System

SooHaeng Yoo,† Dooll Kim, and Sun-Hee Jung*

Structural Biology Center (SBC), Korea Institute of Science & Technology (KIST), 39-1 Hawolgok-dong, Seongbook-goo, Seoul 136-791, Korea
†Dept. of Chemistry, Korea University, Seoul 136-701, Korea

Received June 9, 1999

We varied recombination method of genetic algorithm (GA), *i.e.*, crossover step, to compare efficiency of these methods, and to find more optimum GA method. In one method (A), we select two conformations (parents) to be recombined by systematic combination of lowest energy conformations, and in the other (B), we select them in a ratio proportional to the energy of the conformation. Second variation lies in how to select crossover point. First, we select it randomly (1). Second, we select range of residues where internal energy of the molecule does not vary for more than two residues, select randomly among such regions, and we select either the first (2a) or the second residue (2b) from the N-terminal side, or the first (2c) or the second residue (2d) from the C-terminal side in the selected region for crossover point. Third, we select longest such region, and select such residue (as cases 2) (3a, 3b, 3c or 3d) of the region. These methods were tested in a 2-dimensional lattice system for 8 different sequences (the same ones used by Unger and Moult., 1993). Results show that compared to Unger and Moult's result (UM) which corresponds to B-1 case, our B-1 case performed similarly in overall. There are many cases where our new methods performed better than UM for some different sequences. When cooling factor affecting higher energy conformation to be accepted in Monte Carlo step was reduced, our B-1 and other cases performed better than UM; we found lower energy conformers, and found same energy conformers in a smaller steps. We discuss importance of cooling factor variation in Monte Carlo simulations of protein folding for different proteins. (A) method tends to find the minimum conformer faster than (B) method, and (3) method is superior or at least equal to (1) method.

Introduction

Genetic Algorithms (GA) methods, first introduced by Holland,[1] use the same optimization procedures as natural genetic evolution consisting of mutation, crossover (or recombination) and replication but they operate on strings.[2,3] Recently GA methods have been recognized as an important search technique,[4,5] and applied to various areas such as protein folding,[6-10] RNA folding,[11] flexible ligand docking,[12] etc. In GA, a set or population of current solutions is maintained, which were selected by fitness function for the solutions. The solutions evolve by mutations and crossovers. Mutation is a typical bit change of the string of the solution. Crossover operation lets parts of strings between pairs of solutions exchanged to produce new solutions. This crossover is known to increase the effectiveness of the search, because it allows exploration of regions of the search space inaccessible to either of the two parent solutions.

In protein folding, there is a problem known as the Levinthal paradox;[13] the conformational space for a protein is enormous. If there are three possible backbone conformations per residue, it will leave 3^{100} possible backbone conformations for a 100 residue protein. Several methods have been applied to search the conformational space of a protein for low or global energy conformation, usually Monte Carlo (MC) simulations,[14,15] Genetic Algorithms,[16-20] deterministic[21] or diffusion methods,[22] and other methods.[23]

Application of GA to protein folding problem is in a way

an extension of the conventional Monte Carlo (MC) method to include information exchange between a set of parallel simulations. In this study, GA proceeds as was done by Unger and Moult.[7] A set or population of evolving conformations is maintained, which were selected by fitness function. Each conformation changes independently for some time by the Metropolis MC procedure;[24] this corresponds to the accumulation of point mutations. Then part of two selected polypeptide chains are swapped with each other after a crossover point (crossovers). Metropolis-type criteria are applied to see if each newly generated conformation should be accepted. Those accepted goes through the MC phase again, and the process is iterated.

Implemented in this way, Unger and Moult (abbreviated from here as UM) showed that GA is dramatically superior to conventional MC method for folding on a simple 2-dimensional lattice.[7] We thought that nature would select a special point to crossover (or recombine), and wondered if there's a better way regarding where to crossover besides selecting a crossover point randomly as done by UM. And since in some of GA methods one would select the best fit solutions as two parent solutions to crossover, we tested two different methods of selecting two parent structures for crossover in our GA folding simulation; one selecting the two systematically from the lowest energy pair possible, and the other selecting the two in a probability proportional to its energy which was used by UM. Thus we varied recombination methods of genetic algorithm (GA), *i.e.*, crossover step, to compare the efficiency of these methods, and to find more optimum GA method. These were applied to the same 2-dimensional lattice as used by UM in order to make a comparison with their result. In our study, we also found that different sequences would have the best results on different initial Monte Carlo cooling factor values (which correspond to effective temperature of the system). Therefore we discuss a need to search for and adopt the optimum cooling factor during protein folding simulation for different proteins, because our results suggest that different sequence proteins might have different optimum cooling factor.

Materials and Methods

The Model

Simple model having the essence of the important components of protein folding[25] was adopted, which was used by Unger and Moult;[7] amino acids of only two types, hydrophobic (black; B) and hydrophilic (white; W) are used to represent a linear sequence of protein, and this sequence is folded on a 2-dimensional square lattice where the chain can turn 90°, 180°, and 270° on each lattice point (for lattice model of protein folding simulation, see 26-28). Simple energy function which gives -1 to direct non-diagonal neighboring contacts between non-bonded hydrophobic-hydrophobic amino acids, and gives 0 otherwise was used. 20, 24, 25, 36, 48, 50, 60, and 64 residue sequences specifically used by Unger and Moult (see Table 1) were all used in order to make comparison with their results.

Genetic Algorithm (GA) method

In GA, a population of current conformations is maintained, and the conformations evolve by a number of mutations, executed by conventional Monte Carlo (MC) steps, and crossovers are executed by crossover operations. For each sequence, the number of conformations in the population was 200; we ran MC steps for the total number of residues (*Nres*) and then did crossovers where 200 child (or next generation) conformations were generated for the next population. This makes one generation step, and we stop the simulation either till 100 generations or when the number of minimum energy conformations in the child generation exceeds half of the population, *i.e.*, 100. We start the simulation from all linear or extended conformations.

Monte Carlo (MC) steps of GA. During the MC step, we select randomly a residue to rotate between 2 and (*Nres*-1) for a conformation S_1, then rotate C-terminal portion after the selected residue in a random order among 90°, 180°, or 270°. If we get self-avoiding conformation S_2, calculate its energy E_2. If $E_2 \leq E_1$, then accept the change to conformation S_2. Otherwise, we decide whether to accept the change according to the energy increase with the change, using the following criterion; accept if

$$Rnd < \exp\left[\frac{E_1 - E_2}{c_k}\right]$$

where *Rnd* is a random number selected between 0 and 1, and c_k (cooling factor) is gradually decreased during the simulation by c_k = factor · c_k to achieve a convergence. c_k is kind of effective temperature of the system and was called by cooling factor by Unger and Moult. If the change was not accepted, we retain the former conformation S_1. We repeat this procedure for *Nres* times for each of 200 conformations in the population.

Crossover operations of GA. Crossover or recombina-

Table 1. The following sequences with the number of residues in parenthesis which were tested in UM case were used again to compare the efficiencies of different recombination methods for GA variations. B denotes hydrophobic, and W hydrophilic residues

(20) BWBWWBBWBWWBWBBWWBWB
(24) BBWWBWWBWWBWWBWWBWWBWWBB
(25) WWBWWBBWWWWBBWWWWBBWWWWB
(36) WWWBBWWBBWWWWWBBBBBBBWWBBWWWWBBWWBWW
(48) WWBWWBBWWBBWWWWWBBBBBBBBBBBWWWWWBBWWBBWWBWWBBBBB
(50) BBWBWBWBWBBBBWBWWWBWWWBWWWWBWWWBWWWBWBBBBWBWBWBB
(60) WWBBBWBBBBBBBBWWWBBBBBBBBBBWBWWWBBBBBBBBBBBBWWWWBBBBBBWBBWBW
(64) BBBBBBBBBBBBWBWBWWBBWWBBWWBWWBBWWBBWWBWWBBWWBBWWBWBWBBBBBBBBBBB

tion operation is done in the following way; first select a pair of conformations as parent conformations to be recombined, then select a residue point to crossover or recombine. Then do a crossover where N-terminal portions and C-terminal portions of the parent conformations are swapped and the C-terminal portion is rotated among 0°, 90°, or 270° in a random order until self-avoiding conformation is found for one of the children conformations. If none of six cases leads to self-avoiding conformation, then another pair of conformations is selected. If one self-avoiding conformation is found during three angle trials for one child conformation, then the other child conformation is checked for self-avoiding conformation with the three angle trials. If both child conformations give self-avoiding conformations, the energy of child conformations are calculated and lower energy conformation is selected for the next energy test. When there is just one self-avoiding child conformation, it is used for the energy test. With this conformation S_c and its energy E_c, it is compared to the average energy of parents $\bar{E}_{ij}=(E_i + E_j)/2$; the conformation is accepted if $E_c \leq \bar{E}_{ij}$, and if not, it is accepted if

$$Rnd < \exp\left[\frac{\bar{E}_{ij} - E_c}{gc_k}\right]$$

where gc_{ki} is cooling factor or effective temperature for crossover steps. Again *Rnd* is a random number selected between 0 and 1. This crossover operation is repeated until N-1 newly accepted hybrid conformations have been constructed to make up the population of the next generation. One of the lowest energy conformations (randomly selected if more than one such lowest energy conformations exist) in each generation after the MC steps is directly descended or replicated to the next generation. MC cooling factor c_{ki} was decreased by a factor of 0.97 for every 5 generations, and crossover cooling factor gc_{ki} was decreased by a factor of 0.99 for every 5 generations, just as done by Unger & Moult.[7]

Variations of GA method

To find more optimum GA method, we varied (1) how to select parent conformations, and (2) which residue to select as crossover or recombination point.

Two different methods to select parent conformations. One is (A) to select the pair systematically from the lowest energy pair possible within the population, and the other is (B) to select the pair in a ratio proportional to its energy E_i.

A method:

We realign the conformations within the population in an increasing order of its energy, and we select the following conformations sequentially;

father = $\{S(i)\ i = 1, 2, 3, 4, \ldots\}$

mother = $\{S(j) \mid j = 1, \ldots, i\text{-}1\}$

B method:

The chance $p(S_i)$ of a conformation being selected for crossover is proportional to its energy value in the following way;

$$p(S_i) = \frac{E_i}{\sum_{j=1}^{N} E_j} = \frac{T_i - T_{i-1}}{\sum_{j=1}^{N} E_j}$$

where $T_i = \sum_{p=1}^{i} E_p$ is a partial sum of energies. We select a random number between 0 and 1, then select the conformation such that

father (or mother) = $\{S(i) \mid T_{i-1} \leq Rnd \cdot T_N < T_i\}$

This method is the one used by Unger and Moult.

Three different methods to select crossover point and a subset of them. First, (1) select crossover point randomly among 2 and *Nres* -1. Second, calculate internal energy $e_i[m]$ of the energy of a segment between residue number 1 and residue number m for the conformation S_i, find the regions where at least for three residues these internal energies do not vary, select randomly among such regions, and select either (2a) the first residue, or (2b) the second residue, or (2d) the second from the last residue, or (2c) the last residue of the selected region. These four different crossover points constitute the second group of selecting the crossover point. Third, find the region where these internal energies do not vary for the longest span, and select (3a) the first residue, or (3b) the second residue, or (3d) the second from the last residue, or (3c) the last residue among the span for crossover point. These four different crossover points are the third method of selecting the crossover point. Crossover point is also called recombination point.

Differences in implementation in comparison with Unger and Moult

We used four separate random number generators in our program for random selection processes. While Unger and Moult considered only one child conformation after the crossover, we checked both child conformations generated from crossover to select proper child conformations and selected lower energy conformer of the two for comparison with the average parent energy.

Results

Reason why we tried different Monte Carlo (MC) cooling factors. We tried GA simulation 5 times for each method and for each sequence, as was done by Unger and Moult,[7] and we listed the best results out of 5 trials in Table 2. Unger and Moults GA results (from here will be quoted as UM result) are quoted in Table 3 for comparison, and this corresponds to our B-1 at Monte Carlo (MC) cooling factor c_k = 2.0 case. Table 3 also lists the minimum energy that each sequence would have, which was predicted by UM. Table 2-a shows our simulation results for c_k = 2.0. Compared to UM result, our B-1 at c_k = 2.0 got the same minimum energy conformer for other sequences, but one higher energy conformer for the 50 residue case (-20 instead of -21), and one lower energy conformer for the longest 64 residue case (-38 instead of -37). Therefore in terms of the ability to find the minimum energy conformer, our B-1 case performed similarly to UM result. However, the number of the total steps taken to get to the minimum energy conformer was somehow longer in general for our B-1 case at c_k = 2.0 compared to UM result. As mentioned in previous section, our implementation for B-1 case differs from UM case that we used four separate random number generators for random selection

Table 2(a-e). Results of our GA variation methods. (A) and (B) denote two methods to select parent conformations for recombination, and (1), (2a-d), and (3a-d) denote nine different methods to select crossover point (see Text). The first number in the left columnis the number of residues for each sequence, and the following number in the bracket in the left column is the optimal energy predicted by Ungerand Moult for each sequence (see Table 3). And in the follow 9 columns, the first number shows the lowest energy found in 5 trials of eachmethod; star (*) shows that the method found the lower energy conformer than UM case. The second number in parenthesis is the number of totalGA steps (MC + GA steps) taken to get to the lowest energy conformation. Ck shows the initial value of cooling factor for MC steps.(a) The results at c_k = 2.0 (b) at c_k = 1.5 (c) at c_k = 1.0 (d) at c_k = 0.5 (e) The results of A methods at c_k = 2.5 (upper) and 3.0 (lower).

(a)

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(17.670)	-9(8,753)	-9(17,076)	-9(6,868)	-9(17,753)	-9(12,613)	-9(13,015)	-9(17,820)	-9(21,417)
24< -9>	-9(30.320)	-9(15,609)	-9(20,094)	-9(51,466)	-9(42,182)	-9(25.192)	-9(25,550)	-9(55,544)	-9(10,510)
25< -8>	-8(42.567)	-8(100,890)	-8(80,199)	-8(74,696)	-8(188.267)	-8(112,203)	-8(79,896)	-8(58,150)	-8(58,074)
36<-14>	-14(326,822)	-14(97,901)	-14(281,436)	-14(146,471)	-14(216,792)	-13(97,231)	-14(81,942)	-13(104,767)	-14(231,001)
48<-22>	-22(303,421)	-22(563,949)	-22(245,383)	-22(342,967)	-21(122,454)	-22(326,027)	-22(451,392)	-22(377,399)	*-23(189,353)
50<-21>	-21(965,359)	-21(1,049,842)	-20(635,725)	-21(1,154,789)	-21(1,171,296)	-21(831,734)	-21(636,123)	-21(952,840)	-21(922,299)
60<-34>	-32(249,769)	-34(400,183)	-34(657,331)	-33(226,246)	-34(264,959)	-34(209,787)	-34(235,789)	-33(272,956)	-34(372,582)
64<-42>	*-39(303,251)	*-38(572,950)	*-38(398,999)	-37(519,668)	-37(519,668)	-37(209,787)	*-38(734,524)	*-39(935,110)	*-38(630,400)

residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(25,454)	-9(8,595)	-9(23,368)	-9(34,452)	-9(56,564)	-9(13,543)	-9(8,736)	-9(53,394)	-9(21,561)
24< -9>	-9(157,965)	-9(30,573)	-9(244,138)	-9(170,428)	-9(315,878)	-9(178,225)	-9(111,281)	-9(20,001)	-9(198,124)
25< -8>	-8(248,560)	-8(189,286)	-8(349,792)	-8(312,852)	-8(48,380)	-8(64,313)	-8(415,084)	-8(195,554)	-8(349,687)
36<-14>	-14(569,913)	-14(712,669)	-13(456,373)	-13(718,536)	-14(740,799)	-13(343,715)	-14(373,610)	-14(493,783)	-14(524,380)
48<-22>	-22(567,071)	-22(964,828)	-21(912,878)	*-23(545,097)	-22(1,207,423)	*-23(851,281)	*-23(850,063)	-22(908,881)	-22(884,667)
50<-21>	-20(1,113,638)	-19(927,265)	-20(977,890)	-19(1,091,546)	-19(1,027,631)	-20(1,232,837)	-20(534,526)	-20(789,785)	-19(492,604)
60<-34>	-34(854,069)	-34(1,113,158)	-34(839,881)	-34(809,484)	-34(881,549)	*-35(801,186)	-34(922,834)	-33(1,030,666)	-34(1,093,630)
64<-42>	*-38(1,156,636)	*-38(1,266,707)	*-39(1,154,659)	*-39(1,308,157)	*-38(1,201,391)	*-38(1,179,412)	*-38(1,151,280)	*-38(1,046,657)	-37(1,242,948)

(b)

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(14,074)	-9(13,846)	-9(13,202)	-9(12,912)	-9(18,826)	-9(12,787)	-9(10,083)	-9(8,727)	-9(16,973)
24< -9>	-9(10,619)	-9(15,489)	-9(26,763)	-9(20,159)	-9(20,255)	-9(31,261)	-9(25,489)	-9(25,163)	-9(25,215)
25< -8>	-8(26,770)	-8(48,489)	-8(33,752)	-8(16,289)	-8(21,473)	-8(38,012)	-8(32,100)	-8(28,038)	-8(33,394)
36<-14>	-14(87,641)	-14(127,947)	-13(59,909)	-14(119,276)	-14(208,645)	-14(90,061)	-13(90,346)	-14(84,391)	-13(81,896)
48<-22>	-21(103,837)	-21(114,527)	-21(122,638)	-22(125,255)	-22(207,860)	-22(279,621)	*-23(166,842)	-21(90,558)	-22(159,675)
50<-21>	-21(530,975)	-21(832,247)	-21(514,876)	-21(782,472)	-21(813,598)	-21(573,710)	-21(307,397)	-21(473,959)	-21(450,651)
60<-34>	-34(185,368)	-34(152,314)	-33(114,156)	-34(88,593)	-33(152,287)	-34(188,477)	-33(167,952)	-32(102,261)	-34(343,361)
64<-42>	*-38(146,704)	*-38(243,841)	-37(420,310)	*-38(214,809)	-35(201,488)	*-34(144,440)	*-38(183,848)	*-38(203,720)	-38(282,539)

residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(26,259)	-9(21,551)	-9(8,623)	-9(34,777)	-9(12,506)	-9(4,308)	-9(12,985)	-9(22,139)	-9(30,131)
24< -9>	-9(30,551)	-9(167,994)	-9(40,645)	-9(153,200)	-9(86,728)	-9(51,041)	-9(61,345)	-9(14,393)	-9(80,989)
25< -8>	-8(79,381)	-8(27,343)	-8(171,200)	-8(37,237)	-8(148,610)	-8(16,206)	-8(101,142)	-8(61,039)	-8(525,418)
36<-14>	-14(240,527)	-14(542,362)	-14(293,054)	-14(418,818)	-14(234,430)	-14(345,202)	-14(449,351)	-13(172,736)	-14(134,154)
48<-22>	-21(328,969)	-22(728,956)	-22(677,453)	-22(574,045)	-22(644,430)	-22(568,291)	-22(197,628)	-22(316,364)	-22(565,482)
50<-21>	-21(1,167,552)	-21(1,220,445)	-21(1,107,171)	-20(677,927)	-21(1,181,332)	-21(1,017,028)	-21(1,210,843)	-21(1,127,877)	-21(1,127,419)
60<-34>	-34(570,149)	-34(446,725)	-34(757,894)	-34(424,409)	-34(424,435)	-34(370,316)	-34(889,376)	-33(282,971)	-34(395,295)
64<-42>	*-38(652,023)	*-39(644,628)	*-39(726,363)	*-39(662,449)	*-39(888,754)	*-39(883,315)	*-39(497,031)	*-39(812,716)	-38(524,750)

(c)

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(13,045)	-9(4,655)	-9(12,954)	-9(9,043)	-9(14,604)	-9(4,732)	-9(9,076)	-9(8,760)	-9(12,754)
24< -9>	-9(16,124)	-9(21,265)	-9(15,779)	-9(22,177)	-9(20,614)	-9(10,469)	-9(9,937)	-9(9,953)	-9(15,746)
25< -8>	-8(24,238)	-8(15,697)	-8(16,160)	-8(14,454)	-8(17,994)	-8(16,022)	-8(26,543)	-8(83,912)	-8(15,596)
36<-14>	-13(38,897)	-13(29,750)	-13(37,354)	-13(41,769)	-14(72,338)	-13(37,372)	-14(49,148)	-13(31,720)	-14(75,158)
48<-22>	-22(61,242)	-21(61,033)	-22(63,703)	-21(63,145)	-22(83,572)	-21(69,629)	-21(59,684)	-21(100,704)	-22(114,631)
50<-21>	-21(193,476)	-21(167,796)	-21(173,043)	-21(172,778)	-21(237,463)	-21(135,073)	-2(102,652)	-21(114,641)	-21(113,010)
60<-34>	-33(100,344)	-33(114,796)	-34(88,560)	-33(95,557)	-34(159,673)	-34(100,630)	-34(88,528)	-34(169,062)	-33(106,660)
64<-42>	-35(109,569)	-37(176,910)	-37(81,776)	-37(67,900)	-36(84,146)	-36(190,623)	-36(107,013)	*-38(79,570)	-37(820,707)

Table 2(a-e). Continued

residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(17,054)	-9(16,974)	-9(8,333)	-9(12,661)	-9(13,662)	-9(8,817)	-9(16,981)	-9(4,482)	-9(12,626)
24< -9>	-9(45,653)	-9(35,411)	-9(20,716)	-9(41,194)	-9(35,505)	-9(20,072)	-9(25,133)	-9(40,657)	-9(25,542)
25< -8>	-8(16,025)	-8(32,223)	-8(75,722)	-8(15,625)	-8(26,290)	-8(28,363)	-8(79,919)	-8(21,798)	-8(26,809)
36<-14>	-14(166,430)	-14(91,171)	-14(136,222)	-14(91,727)	-13(53,327)	-14(157,382)	-14(222,382)	-14(105,614)	-14(111,830)
48<-22>	-22(201,445)	-22(321,342)	*-23(153,471)	*-23(217,402)	-22(343,953)	-22(179,837)	-22(189,323)	-22(167,995)	-22(317,130)
50<-21>	-21(612,317)	-21(673,651)	-21(748(895)	-21(209,228)	-21(229,957)	-21(298,819)	-21(225,904)	-21(523,081)	-21(542,059)
60<-34>	-33(521,550)	-34(162,313)	-34(137(328)	-34(187,919)	-33(177,347)	-34(225,889)	-34(337,919)	-34(378,675)	-34(324,456)
64<-42>	*-39(280,573)	*-39(473,285)	*-38(398,784)	-37(238,986)	*-38(279,645)	*-38(369,007)	*-40(262,072)	*-39(644,839)	-38(346,595)

(d)

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(9,826)	-9(4,626)	-9(8,416)	-9(10,462)	-9(13.792)	-9(9,649)	-9(8,418)	-9(9,096)	-9(9,150)
24< -9>	-9(15,411)	-9(16,331)	-9(15,195)	-9(15,392)	-9(13,779)	-9(15,741)	-9(10,683)	-9(10,374)	-9(25,535)
25< -8>	-8(10,338)	-8(10,893)	-8(15,771)	-8(16,104)	-8(10,312)	-8(11,274)	-8(26,641)	-8(11,069)	-8(11,070)
36<-14>	-13(35,659)	-13(31,486)	-13(44,952)	-13(41,765)	-13(31,115)	-13(29,941)	-13(23,829)	-13(17,352)	-13(29,756)
48<-22>	-21(63,130)	-21(64,163)	-21(63,293)	-21(62,870)	-21(71,307)	-22(31,947)	-22(50,732)	-20(39,669)	-21(71,659)
50<-21>	-21(67,355)	-21(105,911)	-21(94,249)	-21(87,815)	-21(97,923)	-21(83,091)	-21(72,369)	-21(63,040)	-21(63,784)
60<-34>	-34(50,291)	-33(52,549)	-33(63,550)	-34(120,853)	-34(93,471)	-34(65,629)	-33(37,064)	-33(64,885)	-33(68,208)
64<-42>	-34(67,695)	-35(274,007)	-35(105,503)	-37(136,290)	-35(96,850)	-35(91,496)	-35(116,270)	-37(258,058)	-37(95,076)

residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(14,512)	-9(15,500)	-9(71,322)	-9(9,293)	-9(12,839)	-9(13,292)	-9(12,848)	-9(8,874)	-9(9,147)
24< -9>	-9(20,999)	-9(15,570)	-9(21,469)	-9(20,695)	-9(26,293)	-9(15,522)	-9(15,641)	-9(23,083)	-9(15,885)
25< -8>	-8(16,263)	-8(16,782)	-8(21,057)	-8(16,735)	-8(31,709)	-8(16,986)	-8(10,310)	-8(157,125)	-8(33,976)
36<-14>	-14(94,896)	-14(105,963)	-13(56,890)	-14(122,115)	-14(38,114)	-13(52,371)	-13(68,678)	-14(164,957)	-14(82,458)
48<-22>	-21(112,743)	-22(151,551)	-22(172,067)	-22(326,567)	-22(271,923)	-21(70,215)	-22(199,667)	-22(139,216)	-21(93,879)
50<-21>	-21(146,971)	-21(167,185)	-21(135,420)	-21(392,217)	-21(299,323)	-21(102,954)	-21(143,973)	-21(113,252)	-21(153,854)
60<-34>	-34(178,620)	-34(154,270)	-34(133,770)	-33(164,480)	-34(162,239)	-34(151,944)	-34(137,619)	-34(184,163)	-34(99,882)
64<-42>	-37(224,976)	*-38(187,736)	*-38(217,914)	-36(159,151)	*-38(218,811)	-37(288,073)	*-39(238,272)	-37(147,055)	*-38(238,982)

(e)

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(31,274)	-9(9,542)	-9(8,725)	-9(12,674)	-9(17,400)	-9(13,354)	-9(21,794)	-9(28,226)	-9(25,668)
24< -9>	-9(35,755)	-9(46,049)	-9(20,048)	-9(89,779)	-9(89,073)	-9(45,944)	-9(40,519)	-9(31,087)	-9(30,427)
25< -8>	-8(282,877)	-8(244,452)	-7(47,336)	-8(394,992)	-8(302,718)	-8(406,192)	-8(228,897)	-8(13,179)	-8(212,963)
36<-14>	-14(255,379)	-14(600,872)	-14(331,127)	-14(499,268)	-14(628,911)	-14(449,731)	-13(239,082)	-14(446,232)	-14(450,298)
48<-22>	*-23(816,243)	*-23(652,626)	*-23(419,119)	-22(805,123)	-22(672,930)	-22(464,566)	-22(731,668)	-22(487,165)	-22(527,678)
50<-21>	-20(691,710)	-20(889,694)	-20(895,661)	-20(1,035,435)	-20(942,931)	-20(689,020)	-20(903,063)	-20(687,343)	-21(1,137,663)
60<-34>	-33(585,080)	-34(708,649)	-33(273,294)	-33(475,037)	-34(737,369)	-34(1,157,085)	-34(494,446)	-33(421,259)	*-35(755,339)
64<-42>	*-38(949,530)	*-39(1,082,667)	*-38(671,181)	*-38(1,108,158)	*-39(916,576)	*-39(1,102,648)	*-38(576,498)	*-41(912,290)	*-39(1,175,757)

residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(15,475)	-9(25,653)	-9(16,983)	-9(10,445)	-9(27,517)	-9(17,656)	-9(8,779)	-9(17,330)	-9(12,916)
24< -9>	-9(20,472)	-9(67,020)	-9(112,726)	-9(160,650)	-9(158,390)	-9(122,228)	-9(55,872)	-9(66,049)	-9(122,128)
25< -8>	-8(154,068)	-8(106,065)	-8(110,922)	-8(216,905)	-8(323,004)	-8(110,922)	-8(531,194)	-8(43,339)	-8(63,147)
36<-14>	-14(647,270)	-13(541,505)	-13(484,687)	-14(741,079)	-14(569,097)	-13(383,276)	-14(682,032)	-14(492,420)	-14(636,943)
48<-22>	-22(1,149,235)	-21(707,495)	-22(1,128,503)	-22(1,215,161)	-22(940,018)	-21(704,953)	-22(1,026,757)	-22(1,002,114)	-21(880,821)
50<-21>	-20(1,105,056)	-20(1,240,480)	-20(1,219,606)	-19(1,187,173)	-19(1,012,227)	-20(647,760)	-20(1,005,735)	-20(800,109)	-20(1,076,968)
60<-34>	-34(693,624)	-34(894,554)	-33(973,239)	-34(1,124,667)	-34(1,071,107)	-33(865,236)	-34(1,168,591)	-33(805,716)	-33(1,084,266)
64<-42>	-37(1,027,489)	-37(921,099)	-36(1,274,701)	*-38(1,023,848)	-37(1,289,619)	-37(1,149,439)	*-38(1,205,773)	-37(1,047,234)	*-38(1,304,191)

Table 3. The results of GA simulation done by Unger and Moult, using the same kinds of sequences. They are quoted from Unger and Moult, J. Mol. Biol. (1993) 231, 75-81. The optimal energies are ones that were predicted by Unger and Moult to be the lowest energy for each sequence. Our new GA recombination methods found the lower energy conformers than the optimal energy for 48 and 60 residue sequences (see Text, Table 2, and Figure 1). The first number in the third column shows the energy of the lowest energy conformer they found, and the second number in parenthesis is the number of the total GA steps (MC + GA steps) taken to get to the lowest energy conformer

Length	Optimal energy	UM
20	-9	-9 (30,492)
24	-9	-9 (30,491)
25	-8	-8 (20,400)
36	-14	-14 (301,339)
48	-22	-22 (126,547)
50	-21	-21 (592,887)
60	-34	-34 (208,781)
64	-42	-37 (187,393)

processes, and that we checked both child conformers after crossover instead of using just one child conformer as was done by UM case.

For the other recombination methods (2a-2d, 3a-3d) and pairing up method (A) at $c_k = 2.0$, we were able to find a lower energy conformer (-39) for the longest sequence of 64 residues, and unexpectedly a lower energy conformer (-23) for the 48 residue sequence and another lower energy conformer (-35) for the 60 residue sequence. In the latter cases, UM predicted that the lowest energy conformer for the 48 residue sequence would be that of the energy -22 and the lowest energy conformer for the 60 residue sequence would be -34, but we found more stable energy conformers than their prediction in both cases. The conformer of the energy -23 for the 48 residue sequence was found in four different methods (A-3d, B-2c, B-3a, and B-3b), while for the 60 residue sequence, the conformer with the energy -35 was found for one method B-3a. The conformations for the energy of -23 and -35 are shown in Figure 1 to show that these conformations are correct and possible.

At $c_k = 2.0$, A-2a, A-3b, and A-3d found all the equal or the better energy conformers compared to UM for all the sequences, but none of B methods were able to find all the minimum energy conformers for all the sequences, especially for the 50 residue case. While looking at the convergency behavior, we found that during the MC step, conformers evolved into higher energy conformer rather easily at $c_k =$ 2.0 for B-1 case. Since larger cooling factor c_k value allows higher energy conformers to be selected more easily during energy comparison process, we thought that the MC cooling factor might not be tight enough to disallow the higher energy conformers during MC step. Therefore we decide to try smaller MC cooling factor c_k value. When we first tried $c_k = 1.0$, results improved for B methods while results got poorer for A-methods. So we also tried $c_k = 1.5$ and 0.5 to find an optimum cooling factor value for different methods.

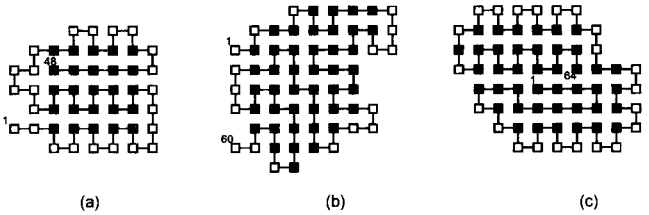

Figure 1. (a) Conformation of the 48 residue sequence showing the energy of -23, (b) that of 60 residue sequence having the energy of -35, both lower than what Unger and Moult predicted to be the lowest energy for each sequence (-22, and -34, respectively) are shown. (c) Confirmation of 64 residue sequence showing the energy of -41, the lowest energy found by our methods is shown for its validity. Denotes hydrophobic residue, and denotes hydrophilic residue. As mentioned in the method section, only non-diagonal direct neighboring contact between non-bonded hydrophobic hydrophobic amino acids gives -1 to the energy and 0 otherwise

We also tried $c_k = 2.5$ and 3.0 for A methods. Simulation results at $c_k = 1.5$ are shown in Table 2-b, those at $c_k = 1.0$ in Table 2-c, those at $c_k = 0.5$ in Table 2-d, and simulation results for A methods at $c_k = 2.5$, and 3.0 are shown in Table 2-e; these are the best results out of 5 trials for each sequence and for each method.

Effect of MC cooling factor. First, we checked the number of cases that each method did not find the same or the better energy conformer compared to UM case for each cooling factor c_k value, and summed up for all the different recombination methods, dividing into only A or B methods of different pairing up scheme. This is shown in Table 4. This shows that for A methods, $c_k = 2.0$ gave the best performance for all the different recombination methods, and for B methods, $c_k = 1.0$ gave the best performance. Notably it was interesting that different pairing up scheme had different dependency on the MC cooling factors.

Table 2-b shows that at $c_k = 1.5$, B methods improved while A method did poorer than $c_k = 2.0$. So for example, among A methods, two methods (A-2c and A-3a) were able to find all the same or the better energy conformers for all the sequences compared to UM case, while among B methods, six methods (B-2a, B-2b, B-2d, B-3a, B-3b and B-3d) were able to find all the same or the better energy conformers for all the sequences. At $c_k = 1.0$ shown in Table 2-c, A methods got poorer while B methods improved than at $c_k =$

Table 4. The number of times that all kinds of differently selecting recombination (crossover) point belonging to either A or B method did not find the same or the better energy conformer compared to UM result, as shown in Table 2, all summed up for all sequences and all (1 to 3d) methods

	A-method	B-method
$c_k = 0.5$	27	8
$c_k = 1.0$	17	3
$c_k = 1.5$	12	4
$c_k = 2.0$	7	14
$c_k = 2.5$	14	
$c_k = 3.0$	20	

1.5 case. None of A methods were able to find all the same or the better energy conformers for all the sequences compared to UM case, while among B methods, seven methods (B-2a, B-2b, B-2c, B-3a, B-3b, B-3c and B-3d) were able to find all the same or the better energy conformers for all the sequences. B-3b method found the conformer with the energy of -40 for the longest sequence of the 64 residues. At c_k = 0.5, both A and B methods got poorer than c_k = 1.0 case, and only three B methods (B-2a, B-2d, and B-3c) were able to find all the same or the better energy conformers for all the sequences compared to UM case. At c_k = 2.5, A methods performed slightly worse in general than at c_k = 2.0 case, and only one A method (A-2d) was able to find all the same or the better energy conformers for all the sequences compared to UM case. However, conformer with the energy of -35 was again found for the 60 residue sequence for A-3d method, and the lowest energy of -41 for the longest 64 residue sequence was found for A-3c method. The conformation of the energy of -41 for the 64 residue sequence is also shown in Table 1 for its validity. In contrast to the overall behavior, results for the longest 64 residue sequence were generally better for A methods at c_k = 2.5 than at c_k = 2.0 case. Simulation results at c_k = 3.0 for A methods shown in Table 2-e are poorer than results at c_k = 2.5.

Thus far we checked only the lowest energy values that each method found, and when we checked the number of the total steps (MC plus GA steps as defined in Materials and Method section) that were taken to get to the minimum energy conformer, we found the following behavior. As the MC cooling factor c_k value decreases (so that it tightens (or decreases) the probability of higher energy conformer accepted during MC evolution processes), Table 2-a to 2-e show that the number of the total steps taken to get to the minimum energy conformer tends to decrease. However for longer sequences, larger MC cooling factor tends to give better results. Therefore there are compensating tendencies for the MC cooling factors which appear to give an optimum MC cooling factor value for all the sequences. In order to find more detailed dependency on the MC cooling factors, we decide to check which MC cooling factor gives the best simulation result for each method and for each sequence, in terms of the lowest energy and the smallest number of the total steps taken to get the lowest energy conformer. This is shown in Table 5 and Table 6. Table 5 lists the best folding simulation result selected among different MC cooling factor c_k cases for each sequence and for each method, and Table 6 lists the value of the cooling factor c_k that gives the Table 5 results, *i.e.*, the best results.

Table 5 shows first that if we choose the best cooling factor c_k result for each sequence and for each method, all the methods perform much better than UM case, finding all the same or the better energy conformers for all the sequences compared to UM case. Also the number of the total steps taken to get to the lowest energy conformer is much better than UM case, and this is also the case for our B-1 case. This shows that our implementation of B-1 method and the other methods are good, and that other methods besides B-1 method appear to be better than B-1 method. If we look at the best

Table 5. The best results among different MC cooling factor ck for each method and for each sequence, in terms of the lowest energy conformer and the smallest number of the total steps taken to get to the lowest energy conformer, are shown in this table. The first number in the left column is the number of residues of each sequence, and the following number in the bracket in the left column is the optimal energy predicted by UM for each sequence. And in the following 9 columns, the first number shows the lowest energy found for each method. The second number in parenthesis is the number of the total GA steps (MC + GA steps) taken to get to the lowest energy conformation, star (*) shows that the method found the lower energy conformer than UM case

residue number <E_{min}>	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20< -9>	-9(9,826)	-9(4,626)	-9(8,416)	-9(6,868)	-9(13,792)	-9(4.732)	-9(8,418)	-9(8,727)	-9(9,150)
24< -9>	-9(10,619)	-9(15,489)	-9(15,195)	-9(15,392)	-9(13,779)	-9(10,496)	-9(9,937)	-9(9,953)	-9(10,510)
25< -8>	-8(10,338)	-8(10,893)	-8(15,771)	-8(14,454)	-8(10,312)	-8(11.274)	-8(26,543)	-8(11,069)	-8(11,070)
36<-14>	-14(87,641)	-14(97,901)	-14 (281,436)	-14 (119,276)	-14(72,338)	-14(90,061)	-14(49,148)	-14(84,391)	-14(75,158)
48<-22>	*-23 (816,243)	*-23 (652,626)	*-23 (419,119)	-22 (125,255)	-22 (83,572)	-22 (31,947)	*-23 (166,842)	-22 (377,399)	*-23 (189,353)
50<-21>	-21 (67,355)	-21 (105,911)	-21 (94,249)	-21 (87,815)	-21 (97,923)	-21 (83,091)	-21 (72,369)	-21 (63,040)	-21 (63,784)
60<-34>	-34 (50,291)	-34 (152,314)	-34 (88,560)	-34 (88,593)	-34 (93,471)	-34 (65,629)	-34 (88,528)	-34 (169,062)	*-35 (755,339)
64<-42>	*-39 (303,251)	*-39 (1,082,667)	*-38 (398,999)	*-38 (214,809)	*-39 (916,576)	*-39 (1,102,648)	*-38 (183,848)	*-41 (912,290)	*-39 (1,175,757)
residue number <E_{min}>	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20< -9>	-9(14,512)	-9(8,595)	-9(8,333)	-9(9,293)	-9(12,506)	-9(4,308)	-9(8,736)	-9(4,482)	-9(9,147)
24< -9>	-9(20,999)	-9(15,570)	-9(21,469)	-9(20,695)	-9(26,293)	-9(15,522)	-9(15,641)	-9(14,939)	-9(15,885)
25< -8>	-8(16,025)	-8(16,782)	-8(21,057)	-8(15,625)	-8(26,290)	-8(16,206)	-8(10,310)	-8(21,798)	-8(26,809)
36<-14>	-14(94,896)	-14(91,171)	-14 (136,222)	-14(91,727)	-14(38,114)	-14 (157,382)	-14 (222,382)	-14 (105,614)	-14(82,458)
48<-22>	-22 (201,445)	-22 (151,551)	*-23 (153,471)	*-23 (217,402)	-22 (271,923)	*-23 (851,281)	*-23 (850,063)	-22 (139,216)	-22 (317,130)
50<-21>	-21 (167,185)	-21 (135,420)	-21 (392,217)	-21 (209,228)	-21 (229,957)	-21 (102,954)	-21 (143,973)	-21 (113,252)	-21 (153,854)
60<-34>	-34 (178,620)	-34 (154,270)	-34 (133,770)	-34 (187,919)	-34 (162,239)	*-35 (801,186)	-34 (137,619)	-34 (184,163)	-34 (99,882)
64<-42>	*-39 (280,573)	*-39 (473,285)	*-39 (726,363)	*-39 (662,449)	*-39 (888,754)	*-39 (883,315)	*-40 (262,072)	*-39 (644,839)	*-38 (238,982)

cooling factor c_k values for each sequence and for each method shown in Table 6, first they differ from each method and from each sequence. Second, for A methods, the best cooling factor c_k value tends to be larger for longer sequences; the 36, 48 and 64 residue sequences tended to prefer larger MC cooling factor value, and the 50 residue preferred smaller MC cooling factor value. For B methods, the tendency is not clear and the shortest 20 residues and the longest 64 residues tended to prefer larger MC cooling factor value among them. Within each method, the best MC cooling factor values for different sequences were all different, which appears to be due to different sequence pattern.

Comparison of recombination methods and pairing up methods. To identify which recombination method is good and which of the two pairing up methods is better, we adopted the following judgement criterion. The results in Table 5, *i.e.*, the best results among different MC cooling factors for each sequence and for each method, were ordered by the lowest energy value they found, and by the smallest number of the total steps taken to get to the lowest energy conformer. This ordering was done separately for the different recombination methods among either A or B method (the best was numbered as 1, and the worst as 9), and also together for all 18 different recombination and pairing up methods (the best numbered as 1 and the worst numbered as 18), for each given sequence. This kind of order among A or B methods, and among all the methods are shown in Table 7 for illustration. For example, for A method and for the 20 residue sequence, A-2a was the best, A-3a the second, A-3c the third, and A-2d the last or the worst among nine recombination method. The order among A methods are shown in Table 7-a, that among B methods in Table 7-b, and the order among all 18 methods are shown in Table 7-c. Also in the table, this kind of the order among different methods for each given sequence was summed up for all the sequences in the next row above the bottom row. The smaller the total sum was, the method was identified with the better performance order in overall at the bottom of each table.

Looking at the Table 7-c, and the Table 5, it is clear that A method tends to find the lowest energy conformers much faster than B method; A methods take up the ranking 1 to the ranking 7 out of 18th order. Therefore this result shows that A method of pairing up is more efficient than B method of pairing up for the crossover operation of the current GA simulation. Among A methods, A-3d is the best in our criterion, followed by A-3b the second, A-1 the third, A-3a and A-3c the fourth best among different recombination methods. For B methods, B-3a is the best, B-3b the second, B-2a the third, and B-3c the fourth, and B-1 ranks the seventh out of nine different recombination methods. These results show that (3) method of locating crossover (or recombination) point in the region where the internal energy from the N-terminal end does not change for the longest residue span is better than or at least equal to (1) method of randomly locating recombination point. With the current results, it is difficult to identify which subset (a to d) of (2) or (3) within each (2) or (3) method of locating recombination point is the best, nor their order in efficiency.

Table 6. The value of the initial MC cooling factor c_k, which gives the best result for each method, *i.e.* results shown in Table 5

residue number	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20	0.5	0.5	0.5	0.5	0.5	1	0.5	1.5	0.5
24	1.5	1.5	0.5	0.5	0.5	1	1	1	2
25	0.5	0.5	0.5	1	0.5	0.5	1	0.5	0.5
36	1.5	2	2	1	1	1.5	1	1.5	1
48	2.5	2.5	2.5	1.5	1	0.5	1.5	2	2
50	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
60	0.5	1.5	1	1.5	0.5	0.5	1	1	2.5
64	2	2.5	2	1.5	2.5	2.5	1.5	2.5	2.5

residue number	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20	0.5	2	1	0.5	1.5	1.5	2	1	0.5
24	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1.5	0.5
25	1	0.5	0.5	1	1	1.5	0.5	1	1
36	0.5	1	1	1	0.5	1	1	1	0.5
48	1	0.5	1	1	0.5	2	2	0.5	1
50	0.5	0.5	0.5	1	1	0.5	0.5	0.5	0.5
60	0.5	0.5	0.5	1	0.5	2	0.5	0.5	0.5
64	1	1	1.5	1.5	1.5	1.5	1	1	0.5

Consistency of results. To check consistency of our results, we ran a whole simulation again for each method and for each sequence, each trying for 5 times to get the best result using different random number starting point. When we order the results as done above and sum the orders as done in Table 7, we get among A methods, the orders of 4 2 8 9 7 2 6 4 1 from A-1 to A-3d with this new simulation results. The orders among B methods for this new set of simulation are 6 4 5 9 3 8 1 6 2 from B-1 to B-3d. These two orders correspond to Table 7-a and 7-b bottom rows which show the orders among different recombination methods in overall. For both A and B methods, the new simulation results give 4 3 6 15 6 1 8 5 2 14 12 13 18 10 16 9 17 10 from A-1 through A-3d, B-1 to B-3d, which corresponds to Table 7-c bottom row. This result shows that the conclusions made in the above are again borne out in the another set of simulation results, and that our results are consistent. In detail, this new set of simulation results also show that A method tends to find the lowest energy conformers much faster than B method. Also they show that (3) method of locating recombination point is better than or at least equal to (1) method of randomly locating recombination point.

Result corresponding to Table 4 for the new set of the simulation gave the similar result, showing that for A methods, $c_k = 2.0$ gave the best performance for all the different recombination methods, and for B methods, $c_k = 1.0$ gave the best performance. In this case also, different pairing up scheme had different dependency on the MC cooling factors. Results corresponding to Table 6 for the new set of the simulation show similar behavior as noted in the above. In detail, the best cooling factor c_k values for each sequence and for each method in this case also differ from each method and from

Table 7. The order among different methods in increasing order of finding the lowest energy conformer in the fastest step, using the results shown in Table 5. (a) shows the order among A methods. (b) that among B methods, and (c) shows the order among both A and B methods (see Text) for each sequence. The sum of the orders among different sequences is made, and the final order among the methods wasmade based on the sum at the bottom row of each table

(a)

residue number	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d
20	8	1	4	3	9	2	4	6	7
24	5	9	7	8	6	3	1	2	4
25	2	3	8	7	1	6	9	4	4
36	5	7	9	8	2	6	1	4	3
48	5	4	3	8	7	6	1	9	2
50	3	9	7	6	8	5	4	1	2
60	2	8	5	6	7	3	4	9	1
64	2	4	9	8	3	5	7	1	6
sum	32	45	52	54	43	36	31	36	29
order	3	7	8	9	6	4	2	4	1

(b)

residue number	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20	9	4	3	7	8	1	5	2	6
24	7	3	8	6	9	2	4	1	5
25	3	5	6	2	8	4	1	7	9
36	5	3	7	4	1	8	9	6	2
48	7	6	1	2	8	4	3	5	9
50	6	3	9	7	8	1	4	2	5
60	7	5	3	9	6	1	4	8	2
64	2	3	6	5	8	7	1	4	9
sum	46	32	43	42	56	28	31	35	47
order	7	3	6	5	9	1	2	4	8

(c)

residue number	A-1	A-2a	A-2b	A-2c	A-2d	A-3a	A-3b	A-3c	A-3d	B-1	B-2a	B-2b	B-2c	B-2d	B-3a	B-3b	B-3c	B-3d
20	15	3	7	5	17	4	7	10	12	18	9	6	14	16	1	10	2	12
24	5	10	8	9	6	3	1	2	4	16	12	17	15	18	11	13	7	14
25	3	4	10	8	1	7	17	5	5	11	13	14	9	16	12	1	15	18
36	7	12	18	14	3	8	2	6	4	11	9	15	10	1	16	17	13	5
48	7	6	5	12	11	10	2	18	3	15	14	1	4	16	9	8	13	17
50	3	10	7	6	8	5	4	1	2	15	12	18	16	17	9	13	11	14
60	3	12	6	7	8	4	5	15	1	16	13	10	18	14	2	11	17	9
64	4	12	18	16	11	23	15	1	14	3	5	8	7	10	9	2	6	17
sum	47	69	79	77	65	54	53	58	45	105	87	89	93	108	69	75	84	106
order	3	7	11	10	6	4	3	5	1	16	13	14	15	18	7	9	12	17

each sequence. Also, for A methods, the best cooling factor c_k value tends to be larger for longer sequences; the 36, 48 and 64 residue sequences tended to prefer larger MC cooling factor value. For B methods, the tendency of preferring larger MC cooling factor c_k value for longer sequences got slightly more clear than previous simulation case. The 24, 25 and 50 residue sequences tended to prefer smaller MC cooling factor value among them. Thus our consistency check shows that the conclusions we can make from our results are reproducible and consistent.

Discussion

Knowing that our implementation of B-1 method and the other methods is proper, and gives better performance than Unger and Moults result (UM), we could make the following conclusions and suggestions. First, A method of systematically selecting the lowest energy pair possible for crossover tends to find the lowest energy conformer faster than B method of selecting the parent with a probability proportional to its energy. Second, (3) method of locating crossover or recombination point in the region where the internal energy from the N-terminal end does not change for the longest residue span is better than or at least equal to (1) method of randomly locating crossover point in their performance.

Third, within each method, the best MC cooling factor values for different sequences were all different. This result make us reminded importantly that depending on the nature of sequence context (hydrophobic and hydrophilic) and pattern, folding pathway will be different, especially all the energy bumps and transition states (in terms of their location and height) that it has to go through will be different. Therefore, the MC cooling factors which affects the size of energy bumps that folding process passes through will guide folding pathway of each sequence differently during the simulation. Therefore our results suggest that for different sequences, the best MC cooling factor value will be different, and for each sequence, we have to search and use different MC cooling factor value during protein folding simulation. This is one of the most important findings of our results. We therefore suggest that for protein folding simulation, we should optimize the MC cooling factor for each different sequence and use different optimum MC cooling factor for different protein sequence for their best and optimum performance. Even though our results are obtained from GA simulation, we think that this suggestion could apply to either conventional MC or other kinds of (protein or RNA) folding simulation because we varied the MC cooling factor values. This dependency on the MC cooling factors might have corresponding meaning to the fact that different proteins do have

different optimum temperature range where they refold well in solution.

Fourth, the region where the internal energy from the N-terminal end does not change for more than two residues (*i.e.* at least three residues) does not have any hydrophobic-hydrophobic contact with the rest of the molecule, and can be thought of (protruding) loops (or coils) in their secondary structure representation for the 2-dimensional lattice system. Our results suggest that it is better to recombine (or cross-over) two parent structures in this kind of loop region for the crossover operation of the GA protein folding simulation rather than to select randomly a point which could fall into a region of the α-helix or β-sheet where hydrophobic-hydrophobic contact might exist.

Acknowledgment. We thank Dr, B. K. Lee at NCI, NIH for giving us an idea for varying one of the recombination method, and Prof. Sung-Hou Kim at Berkeley for giving us encouragement all along. We also thank prof. Seung-Joon Jeon at Dept. of Chemistry, Korea University for guidance of Mr. Yoo.

References

1. Holland, J. H. *Adaptation in Natural and Artificial Systems*; The Univ. of Michigan Press: Ann Arbor, MI, 1975.
2. Holland, J. H. *Scientific American* **1992**, *July*, 44.
3. Forrest, S. *Science* **1993**, *261*, 872.
4. Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, 1989.
5. Davidor, T. *Genetic Algorithms and Robotics: A Heuristic Strategy of Optimization*; World Scientific: New Jersey, 1990.
6. Schulze-Kremer, S. *Parallel Problem Solving from Nature 2*; Manner, R., Manderick, B., Eds.; Elsvier Science Publishers B.V.: 1992; p 391.
7. Unger, R.; Moult, J. *J. Mol. Biol.* **1993**, *231*, 75.
8. Sun, S. *Protein Science* **1993**, *2*, 762.
9. Dandekar, T.; Argos, P. *J. Mol. Biol.* **1994**, *236*, 844.
10. Perdersen, J. T.; Moult, J. *Proteins: Struct., Funct. Genetics* **1995**, *23*, 454.
11. Van Batenburg, F. H. D.; Gultyaev, A. P.; Pleij, C. W. *J. theor. Biol.* **1995**, *174*, 269.
12. Oshiro, C. M.; Kuntz, I. D.; Dixon, J. S. *J. Comput.-Aided Mol. Design* **1995**, *9*, 113.
13. Levinthal, C. *J. Chim. Phys.* **1968**, *65*, 44.
14. Rooman, M.; Kocher, J.-P.; Wodak, S. *J. Mol. Biol.* **1991**, *221*, 961.
15. Kang, H. S.; Kurochkina, N. A.; Lee, B. *J. Mol. Biol.* **1993**, *229*, 448.
16. Elofsson, A.; Le Grand, S. M.; Eisenberg, D. *Proteins: Struct., Funct. Genetics* **1995**, *23*, 73.
17. Sun, S.; Thomas, P. D.; Dill, K. A. *Prot. Engineering* **1995**, *8*, 769.
18. Perdersen, J. T.; Moult, J. *J. Mol. Biol.* **1997**, *269*, 240.
19. Dandekar, T.; Argos, P. *Protein Engineering* **1997**, *10*, 877.
20. Bailey, M. J.; Jones, G.; Willett, P.; Williamson, M. P. *Protein Science* **1998**, *7*, 491.
21. Samorjai, R. L. *J. Phys. Chem.* **1991**, *95*, 4141.
22. Piela, L.; Kostrowicki, J.; Scheraga, H. A. *J. Phys. Chem.* **1989**, *93*, 3339.
23. Srinivasan, R.; Rose, G. D. *Proteins: Struct., Funct. Genetics* **1995**, *22*, 81.
24. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.
25. Dill, K. A. *Biochemistry* **1990**, *29*, 7133.
26. Covell, D. G.; Jernigan, R. L. *Biochemistry* **1990**, *29*, 3287.
27. Skolnick, J.; Kolinsky, A. *Science* **1990**, *250*, 1121.
28. Shakhnovich, E.; Fartdinov, G.; Gutin, A. M.; Karplus, M. *Phys. Rev. Let.* **1991**, *67*, 1665.