DNA Sequencing by Oligonucleotide Hybridization: A Genetic Algorithm Approach

Chinmay Majee[1], G Sahoo[2]
Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India, Mesra-835215
[1]Sfzszz96@yahoo.co.in , [2]Drgsahoo@yahoo.com

Abstract. In this paper we examine the performance of Genetic Algorithm in DNA Sequencing through Oligonucleotide Hybridization Method. We construct special kind of Crossover and Mutation operators. Moreover we introduce a special kind of scoring scheme for the hybridized probes for detecting the repetitive units in the DNA chain. We also reconstruct probes of length (n+1) from the hybridized probes of length n in the original biochemical experiment. Furthermore, we apply GA simultaneously to the two populations consisting of DNA sequences of arbitrary length formed from the probes of length n and (n+1) respectively. Finally we compare the results of the experiment and analyze the performance of GA in the DNA sequencing

1 Introduction

DNA molecule- the fundamental fabric of life consists of a sugar-phosphate backbone and attached to it, a long sequence of four kinds of nucleotide bases. At an abstract level a single DNA strand can be approximated as a string over the alphabet [A, T, G, C] which are the nucleotide bases. Sequencing a DNA molecule means determining the sequence of the nucleotide bases (A, T, G, C) in the long DNA chain.

In the current laboratory methods of DNA sequencing (like Maxam Gilbert or Sanger method),a long DNA strand (for example a human DNA can consist 3×10^9 base pairs) is chopped into fragments consisting of 200 to 500 base pairs and then biochemical method is applied to detect the nucleotide sequence in a single fragment. Finally these known fragments are superposed to form a superstring which approximates the original DNA strand. The conventional DNA sequencing problem can be cast as a Minimum Length Superstring Problem for which there exists some approximate algorithms which approximate the shortest superstring of length n by a superstring of length O(n log n)[1].

However these methods are costly and involve number of biochemical experiments. Moreover, superposition of these fragments is a random procedure (since there is no knowledge available about the site of the original strand from which the fragment has been chopped) and may not produce the exact DNA strand.

These limitations of the conventional DNA sequencing methods can be overcome in the oligonucleotide hybridization technique; where in a single biochemical experiment (using a DNA probe chip) it is possible to know almost all the sub-sequences of a specified length which are present in the original string. The success of hybridiza-

tion technique lies in developing an efficient algorithm to reconstruct the original strand from the available information regarding the sub-sequences.

Genetic Algorithm has been applied to sequence the DNA molecule through Hybridization technique by Douzono, Hara and Noguchi [2].

However, the main problem lies in detecting the repetitive units in the sequence. In our work we have introduced a special scoring scheme based on the matching of probes to detect the repetitive units in the sequence. Moreover, we have reconstructed probes (or sub-sequences) of length n+1 from the original set of probes of length n obtained from the hybridization experiment. We have constructed two populations consisting of sequences formed from the probes of length n and n+1 respectively. GA was applied simultaneously to both of these populations. We have also constructed special kind of crossover and mutation operator involving the connection rule. Though these two populations evolve almost independently, we have allowed inter-population crossover between these two populations. For short sequences our simulation reconstructed almost the exact DNA sequence.

2 DNA Chip and Hybridization

A DNA probe chip is a biochemical chip which contains all possible combinations of sub-sequences or probes (of a specified length N) of nucleotide bases. Hence total number of probes in a DNA probe chip will be 4^N. For example, if N equals 4, then the probe chip will contain all possible probes of length 4 starting from AAAA to CCCC. The probes on the chips are built by combinatorial chemical synthesis [3].

In hybridization technique, a solution containing the DNA fragment that will be sequenced is washed over the DNA probe chip. The sub-sequences of the DNA strand will hybridize the complementary probes (A-T and G-C are complementary to each other) in the DNA probe chip. For, example if the original DNA fragment is GACCATCGGA, then the hybridized probes will be GTAG(CATC), TAGC(ATCG), TGGT(ACCA), CTGG(GACC), GGTA(CCAT), GCCT(CGGA), AGCC(TCGG). The strings within the bracket represent the sub-sequences present in the original string. The hybridized probes have been given here in a random sequence to show that the order information of the probes will not be available from the hybridization experiment.

In a way, the oligonucleotide hybridization technique involves single biochemical experiment in contrast with the conventional methods of DNA sequencing. However, the main problem in hybridization technique is to reconstruct the target DNA sequence from the information available from the hybridized probes.

3 Application of GA

Genetic Algorithms are the search algorithms which simulates natural evolution in optimization and search. Since, sequencing the DNA molecule involves massive compu-

tation and reconstructing the original DNA sequence is an optimization search in the domain of different sequences those can be constructed with the hybridized probes hence application of GA is quite justified in the DNA sequencing problem.

The algorithm that we have proposed for the DNA sequencing problem through oligonucleotide hybridization technique is discussed below.

3.1 Reconstruction of Sub-sequences of Length N+1

In the actual hybridization experiment the probes of length N are obtained. From those probes of length N, the sub-sequences of length N+1 are constructed using a special kind of connection rule [3].

The connection rule is explained below. Suppose the hybridized probes are GGTA, TGGT, GTAG, AGCC, GCCT, TAGC, CTGG. Here we have taken N = 4. Now if any two probes are found such that the last three (N-1) letters (nucleotides) of one probe occur in the beginning of the other probe then these two probes can be joined to form a sub-sequence of length N+1(in this case N+1 = 5).For example, the last three letters of GTAG occur in the beginning of TAGC. Hence, GTAG and TAGC can be joined to form GTAGC which is of length 5. Similarly, TAGC and AGCC can be joined to form TAGCC.

It can be proved that all the sub-sequences of length N+1 those can be constructed from the probes of length N are the valid sub-sequences, that is, they actually occur in the original DNA strand. However, it can be observed that if one constructs further sub-sequences of length N+2 from the sub-sequences of length N+1, those may not be valid sub-sequences. Moreover, construction of sub-sequences of length lesser than N from the hybridized probes will not provide any useful information as they are already embedded in the probes of length N. Hence, N+1 is the optimal length of the sub-sequences which can be constructed from the probes of length N. In our simulation, we have observed that in many cases population constructed from the sub-sequences of length N+1 has produced better result than the population constructed from the probes of length N.

3.2 The Scoring Scheme of the Probes

In nature, almost all DNA sequences have repetitive sequences. Hence, the main problem lies in detecting these repetitive sequences. To detect these repetitive units we have introduced a scoring scheme for the probes and sub-sequences. Before, explaining the scoring scheme, let us analyze the informal basis of the scoring scheme.

Let us consider a sequence GTAGCCGTAGGCGTAG. This sequence contains sub-sequences GTAG, TAGC, AGCC, GCCG, CCGT, CGTA, GTAG, TAGG, AGGC, GGCG, GCGT, CGTA, GTAG given in the sequential order. However the sub-sequences GTAG and CGTA are repeated thrice and twice respectively. If an identical DNA sequence were hybridized in the hybridization experiment the hybridized probes obtained from the experiment would be complementary to the sub-

sequences GTAG, TAGC, GCCG, CCGT, CGTA, AGCC, AGGC, TAGG, GCGT, GGCG. The order of the sub-sequences has been altered and the repetitive units have been omitted.

Now, if GTAG is matched with other sub-sequences, it would be found that the first letter of GTAG, the G has occurred at the end of other sub-sequences thrice (GCCG, TAGG, GGCG). Out of these three sub-sequences GCCG and GGCG actually superpose with the sub-sequence GTAG to form valid sub-sequences of the original sequence (GCC*G*TAG, GGC*G*TAG). The letter in the italics in the sub-sequences within the bracket is the overlapping letter (i.e. nucleotide) of the two units. Similarly first two letters of GTAG, GT occur at the end of other sub-sequences only twice (CCGT, GCGT). Both of these sub-sequences join with GTAG to form valid sub-sequences CC*GT*AG and GC*GT*AG respectively. The first three letters of GTAG, GTA occur in the end of the other sub-sequences only once (CGTA) and the corresponding valid sub-sequence is C*GTA*G. The last three letters of GTAG, TAG occurs in the beginning of other sub-sequences twice (TAGC, TAGG). GTAG superposes with both of these sub-sequences to form the valid sub-sequences (G*TAG*C and G*TAG*G respectively. The last two letters of GTAG, AG occur in the beginning of other sub-sequences twice (AGCC, AGGC). GTAG joins with both of these.sub-sequences to form valid sub-sequences GT*AG*CC and GT*AG*GC. The last letter of GTAG, the G occurs in the beginning of other sub-sequences thrice (GGCG, GCCG, GCGT). GTAG combines with only two of these sub-sequences (GGCG, GCCG) to form valid sub-sequences of the string (GTA*G*GCG, GTA*G*CCG).

Similarly, if each sub-sequence is matched with other sub-sequences in the set and a score is assigned to each sub-sequence based on the number of matches, it can be observed that more the sub-sequence has been repeated in the original sequence, higher is its chance of sharing a common unit with other sub-sequences. For example, in the above example, since GTAG has been repeated thrice in the original sequence, it will have more number of matches with other sub-sequences compared to any other sub-sequence. Hence in a scoring scheme based on number of matches, GTAG will have more score compared to other sub-sequences. However, it can be observed that if the length of the matching unit (the common letters between two sub-sequences) is small (for example a single letter), the number of matches will be more (for example, for the first letter of GTAG, G, there are 3 matches out of which only two can produce valid sub-sequences) and all of these matches may not produce valid sub-sequences. However if the length of the matching unit becomes large, then the number of matches may be small, but there is high probability that all these matches will produce valid sub-sequences. For example, for the last three letters of GTAG, TAG, there are two matches. But both these matches correspond to valid sub-sequences of the string. Hence a successful scoring scheme will try to bias the score towards matches involving long matching units. In our scoring scheme we incorporate this concept by introducing a normalized scoring scheme. It can also be observed that if there is a repetitive chain of a single unit (like AGCC*AGCC*AGCCAGCC where the single unit AGCC has been repeated four times) then, not much knowledge will be available regarding the number of repeats in the sequence. However, in our scoring scheme we have taken into account double consecutive repetition of a single unit. From the above discussion it is evident, that repetition is not an isolated incident.

Knowledge about repetition can be obtained, by carefully observing the relation of one sub-sequence with other sub-sequences in the set.

Now we will explain the scoring scheme adopted for the simulation. In the example discussed above, the hybridized probes were complementary to the sub-sequences GTAG, TAGC, GCCG, CCGT, CGTA, AGCC, AGGC, TAGG, GCGT, GGCG. Henceforth in our discussion, we will refer to these sub-sequences rather than the actual hybridized probes (which are complementary to these sub-sequences) for clarity of understanding. For each sub-sequence S_i of length N, we associate a 2(N-1) dimensional vector A_i. Now let us take the sub-sequence S_6 which is AGCC (the sub-sequences are numbered in the order they are given here). The first letter of AGCC, A occurs in the end of other probes only once (CGT*A*). The first two letters of AGCC, AG occur in the end of other sub-sequences only once (GT*AG*). The first three letters of AGCC, AGC occur in the end of other sub-sequences only once (T*AGC*). The last three letters of AGCC, GCC occur in the beginning of other sub-sequences only once (*GCC*G). The last two letters of AGCC, CC never occur in the beginning of any other sub-sequence. The last letter of AGCC, C occurs in the beginning of other sub-sequences only once (*C*CGT). Hence corresponding vector A_6 can be constructed as [1 1 1 1 0 1]^T. Similarly A_2 (for the sub-sequence TAGC) can be constructed as [2 1 1 1 1 2]^T. Vectors for other sub-sequences can be constructed in similar way on the basis of matching. After the construction of the match-vectors, each sub-sequence is checked for double consecutive repetition. For example if AGCC has been repeated consecutively in a sequence (that is the sequence contains a substring AGCCAGCC), then the sub-sequences AGCC, GCCA, CCAG, CAGC will also occur in the set. Hence, for AGCC, it is checked whether these sub-sequences are present in the set of complementary sub-sequences of the original hybridized probes. If these sub-sequences are found then only AGCC has a double consecutive repetition in the sequence.

Now score against each of these sub-sequences can be calculated as follows: score for sub-sequence A_i,

$$P_i = \left[\left(\left(\sum_{j=1}^{2(N-1)} \left(\frac{A_i[j]}{\sum_{i=1}^{N1} A_i[j]} \right) \right) + 0.5 \times d_i \right) \times \text{int} \left[\left(\frac{dnalow + dnahigh}{2} \right) - 3 \right] \right] . \qquad \textbf{(1)}$$

Here $A_i[j]$ denotes the j^{th} element of the vector A_i. $N1$ is the number of sub-sequences in the set. d_i is a number which takes binary values 0 and 1. If, the sub-sequence S_i has double consecutive repetition, then d_i = 1 else it is 0. *dnalow* and *dnahigh* are the lower and upper limits of the length of the sequence. Since the exact length of the DNA sequence is not known, a range is assumed by providing the two

limits *dnalow* and *dnahigh*. It can be observed that the term $\left(\sum_{j=1}^{2(N-1)} \left(\frac{A_i[j]}{\sum_{i=1}^{N1} A_i[j]} \right) \right)$ is the normalized score based on the matching.

In our simulation we have constructed the sub-sequences of length (N+1) from the hybridized probes of length N. The same scoring scheme was also applied to the sub-sequences of length N+1.

3.3 The Population Construction

In the simulation two populations were used. One population was constructed from the hybridized probes of length N (actually from the complementary sub-sequences of the hybridized probes) and the other was constructed from the sub-sequences of length N+1.

Each population consists of chromosomes of arbitrary length formed from the sub-sequences by connection rule. The chromosome formation is done as follows: suppose the set of sub-sequences of length 4 is (GTAG, TAGC, GCCG, CCGT, CGTA, AGCC, AGGC, TAGG, GCGT, GGCG).An arbitrary length is chosen for the chromosome using the random number generator. Now a sub-sequence is randomly selected from the set of subsequences. (Let this sub-sequence be TAGG). This sub-sequence is matched with other sub-sequences. If some sub-sequences are found from the set which can be connected to the sub-sequence following the connection rule, then any one of these sub-sequences is randomly selected and the sub-sequence under consideration is connected to this sub-sequence following connection rule. For example, T*AGG* can be connected to *AGG*C to form T*AGG*C. This procedure goes on until the length of the chromosome does not cross the chosen arbitrary length. For example TAGGCG can be a chromosome of length 6 constructed by the above procedure.

The same procedure is followed for constructing the second population where each chromosome is constructed from the sub-sequences of length N+1 following the connection rule.

3.4 Fitness Function

The fitness function for a chromosome is calculated as the sum of the scores of the sub-sequences present in the population.

3.5 Selection and Crossover

In our simulation we have adopted roulette wheel selection method for the chromosomes. However, the crossover operator was specially constructed using the connection rule

We have used two types of crossover: intra-population crossover and inter-population crossover. In intra-population crossover two chromosomes of the same population take part. In inter-population crossover one chromosome from first population and another chromosome from second population take part

Crossover operators follow the connection rule. For intra-population crossover, two arbitrary chromosomes are selected from any population. Two chromosomes are matched to find common sub-sequences of length N-1 (if the two chromosomes come from first population) or N (if the two chromosomes come from second population) so that they can be swapped using connection rule. If more than one common site are found, then any one of them is randomly selected. Then both the chromosomes exchange their letters (nucleotides) using connection rule. For example, let two chromosomes from second population be *GCAGCTA*GCCAGGCGT and TACT*GCTA*AGG*GCAG*C (we have taken N=4). Now both these chromosomes have common substrings GCTA and GCAG of length 4. Now randomly one sub-string is chosen (let it be GCAG).Then the intra-population crossover between these two strings will produce two strings GCAGC and TACTGCTAAGGGCAGCTAGCC AGGCGT. If however, GCTA were chosen as the common string instead of GCAG, then resultant substrings would be GCAGCTAAGGGCAGC and TACTGCTAGCCAGGCGT respectively. It can be observed that if the two crossover sites are at more or less equal position in the two strings then the homogeneity of length in the resultant chromosomes is conserved. But if the two crossover sites occur at different positions in the strings then, two resultant chromosomes vary widely in their length.

For inter-population crossover also, the crossover operator is constructed following the connection rule. Now suppose, GCAGCTAGCCAGGCGT comes from first population and the string TACTGCTAAGGGCAGC comes from second population. Since the first chromosome comes from first population, hence the resultant string of the inter-population crossover will belong to the first population. Hence, for connection rule, we will search for the common substrings of length N-1(i.e.3).The common substrings of length 3 are GCT, CTA, CAG, AGC. If the substring CTA is randomly chosen then, the resultant chromosome will be GCAGCTAAGGGCAGC (only the first resultant string is taken, since it will go to the population from which the first chromosome of the two parent chromosomes comes).

3.6 Mutation Operator

Mutation operator also follows connection rule. We have introduced three types of mutation operator: 1.Break-mutation 2.Add-mutation 3.Del-mutation.

In break-mutation, the chromosome is broken at a random site. Then, two arbitrary lengths are set for two broken fragments. Then the set of sub-sequences are searched, so that the length of each broken segment can be increased by joining sub-sequences from the set using connection rule until the length of the string does not cross the chosen length. Then, the fragment which has more fitness value is chosen as the mutated replacement of the original chromosome. For example, let the chromosome be CGT*A*GCCG and the set of sub-sequences of length 4 be (GTAG, TAGC, GCCG,

CCGT, CGTA, AGCC, AGGC, TAGG, GCGT, GGCG). Now, the chromosome is broken at a random site (shown in italics) and the two fragments are CGTA and GCCG. Let the chosen lengths for these two fragments are 6 and 7. Now, the set of sub-sequences is searched for increasing the length of these two fragments. The full grown fragments are CGTAGC and GCCGTAG. Of these two sequences, whichever has higher fitness value, would be selected as the mutated replacement of CGTAGCCG. However, during growth of the fragments, if at a point it is found that there is no sub-sequence available in the set which can be joined to the fragment following the connection rule, then the growth process is aborted automatically.

In add-mutation the, the chromosome is allowed to grow by joining to it the sub-sequences from the set using the connection rule, until its length crosses a randomly chosen length.

In del-mutation a part of the chromosome is deleted using the random number generator.

4 Simulation Results

For short sequences the simulation produced near exact results. The DNA sequences of different lengths were fed to the machine randomly and the program produced hybridized probes (actually the complementary probes of length n = 4) and constructed the sub-sequences of length n+1(i.e. 5). It also calculated a range of the length of the sequence in terms of lower limit of the length of the DNA sequence (*dnalow*) and the upper limit (*dnahigh*) of the length of the DNA sequence. Also we calculated similarity index (S-index) for each generation which gives a measure of the similarity between the best sequence of that generation (the best sequence has the highest fitness value between all the sequences of first and second population of a particular generation) and the original DNA sequence. If the number of the positions where the two sequences differ is q and length of the original DNA sequence is n then S-index = $(1-q/n) \times 100$.

In the simulation, if crossover occurs (with a probability pcross1 for first population and probability pcross2 for second population), then first intra-population crossover will be tried(with a probability pintracross1 for the first population and a probability pintracross2 for second population).If intra-population crossover does not occur, then inter-population crossover will be tried (with a probability pintercross1 for first population and a probability pintercross2 for second population). Similarly in case of mutation, if mutation occurs (with a probability pmute1 for first population and a probability pmute2 for second population), then break-mutation will be tried first (with a probability pmutbreak1 for first population and a probability pmutbreak2 for second population). If break-mutation does not occur, then add-mutation will be tried next (with a probability pmutadd1 for first population and a probability pmutadd2 for second population).If add-mutation does not occur then del-mutation will be tried next (with a probability pmutdel1 for first population and pmutdel2 for second population). Hence, the effective crossover and mutation probabilities will differ from the probabilities those will be shown here.

Below we present simulation results for two short DNA sequences.

Sequence 1:

Original sequence: GCTTATGCGTCA

Length of the sequence = 12

No. of probes (actually complementary sub-sequences) of length 4 (we have taken N=4) constructed = 8.

No. of sub-sequences of length N+1 (i.e. 5) constructed = 7

Population size = 20 (This is same for both the populations)

dnalow = 11 *dnahigh* = 13

No. of generations = 8

pcross1 = 0.7 pintracross1 = 0.7 pcross2 = 0.7 pintracross2 = 0.9

pmute1 = 0.2 pmutbreak1 = 0.006 pmutadd1 = 0.6

pmute2 = 0.2 pmutbreak2 = 0.002 pmutadd2 = 0.9

The variation of maximum fitness value (max1), minimum fitness value (min1) and the mean fitness value (avg1) of first population with the generations is shown below:

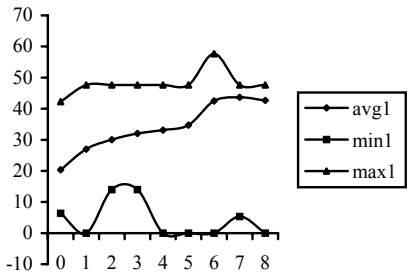

Fig. 1. Variation of maximum, minimum and mean fitness values with generations for the first population for sequence 1 (no. of generation is plotted along x-axis and the fitness value is plotted along y-axis)

The variation of maximum fitness value, minimum fitness value and the mean fitness values with the generations for the second population is shown below:

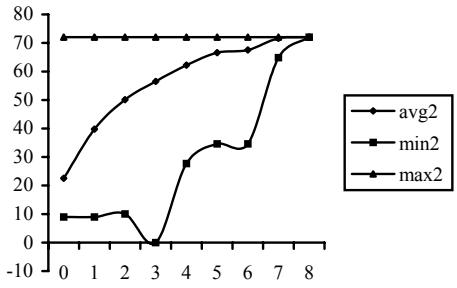

Fig. 2. variation of maximum, minimum and mean fitness values with generation for the second population for sequence 1

The similarity index remained almost same (91.66) throughout the generations. It can be observed that for both the populations the maximum fitness value does not vary sharply, while the minimum fitness value varies sharply. The average fitness steadily grows up. The final sequence that was selected after the 8^{th} generation is GCTTATGCGTC (this sequence has the highest fitness value of all) which is almost identical with the original sequence except that it has length 11.

Sequence 2:
Original sequence: GTTGGAGTCGCAGCTGCCAA
Length of the sequence = 20
No. of probes (actually the complementary sub-sequences) of length 4 constructed = 16
No. of sub-sequences of length 5 constructed = 15
Size of the population = 8
dnalow = 18 *dnahigh* = 22
No. of generations = 30
pcross1 = 0.9 pintracross1 = 0.6 pcross2 = 0.9 pintracross2 = 0.6
pmute1 = 0.2 pmutbreak1 = 0.02 pmutadd1 = 0.9
pmute2 = 0.2 pmutbreak2 = 0.02 pmutadd2 = 0.9

The variation of maximum, minimum and mean fitness values of the first population with generations is shown in Fig. 3:

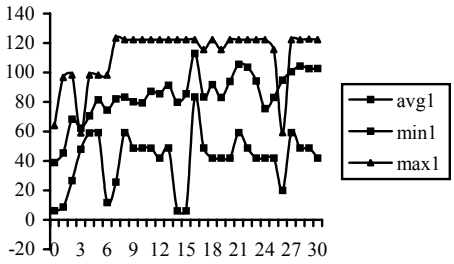

Fig. 3. Variation of maximum, minimum and mean fitness values with generations for first population for sequence 2

The variation of maximum, minimum and mean fitness values of second population with the generations is shown below.

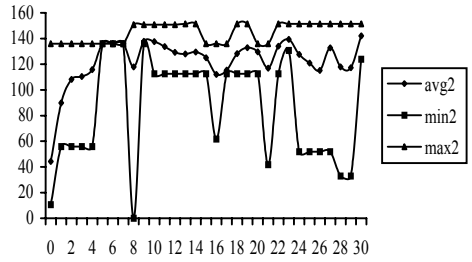

Fig. 4. Variation of maximum, minimum and mean fitness values of second population with generations for sequence 2 (x-axis values are no. of generations and y-axis values are fitness values).

It can be observed that for both the populations the maximum fitness value does not vary sharply. The reason for this is probably the early convergence of the populations. However the minimum fitness value and the mean fitness value vary sharply. Similarity index was high in the beginning (95.00), but after some early variations,

became steady (59.99). The high S-index in the early generations is accidental, and its fall in some later generations reflects the actual variation of the S-index

Although, the S-index remains almost constant at 59.99 in the later generations, the sequence with highest fitness value after generation 30 is:
GTTGGAGTCGCAGCTGGAGTCG (length = 22) which is almost identical (except the last few letters) with the original sequence.

5 Conclusion

In our work we have applied GA in the sequencing of DNA molecules by oligonucleotide hybridization method. We have introduced special scoring scheme for detecting the repetitive units. We have constructed sub-sequences of length N+1 from hybridized probes of length N actually available from the bio-chemical experiment.

We have constructed two populations which consist of sequences formed from the sub-sequences of length N and N+1 respectively. We have constructed special kind of crossover and mutation operator based on connection rule. For short sequences, our algorithm produced good results. However, our algorithm suffered from the early convergence of the population which, we believe can be corrected with proper modification of our algorithm. For long sequences, we expect our algorithm shall produce satisfactory results.

References

1. Ming Li: "Towards a DNA Sequencing Theory (Learning a String)", CH2925-6/90/0000/0125 1990 IEEE 125-134
2. Hiroshi Douzono, Shigeomi Hara, Yoshi Noguchi: "An Application of Genetic Algorithm to DNA Sequencing by Oligonucleotide Hybridization", 0-8186-8548-4/98 1998 IEEE 92-98
3. Gary B. Fogel, Kumar Chellapilla: "Simulated Sequencing by Hybridization Using Evolutionary Programming", 0-7803-5536-9/99 1999 IEEE 463-469