3rd International Conference on Recent Development in Engineering Science, Humanities and Management
National Institute of Technical Teachers Training & Research, Chandigarh, India (ESHM-17)
26th March 2017, www.conferenceworld.in ISBN: 978-93-86171-35-1

COMPUTING WITH DNA

Prof. J.P. Singh

Prof , Jagan Institute of Management Studies-Rohini, GGSIPU, Delhi-85

Adds-Q-551 Rishi Nagar, Rani Bagh, Delhi (India)

ABSTRACT

DNA computing [1] is a new research area which has been receiving more and more attentions from both biologists and computer scientists. It is a new computation paradigm which propose the use of molecular biology tools to solve mathematical problems. By this computation we would able to solve NP-complete.

Hamiltonian – cycle problem , traveling-salesman problem and subset-sum problem can be solved by DNA computing method. I am using this technique to solve the Direct Hamilton path Problem[2], The Traveling-Salesman problem[3] , Conjunctive Normal Form(CNF) problem[4] and Knapsack problem[17]. Traveling – Salesperson Problem is an instance of optimization problems on weighted graphs . For any weighted graph G=(V,E), $v_i \in V$, $1 \le i \le n$, *where exists weight* w_{ij} *on edge* $v_i v_j$, *we use two DNA strands with different lengths to encode each of the edges. This work extends the capability of DNA computing to solve numerical optimization problems.*

I. INTRODUCTION

DNA computing is a new computation paradigm which proposes the use of molecular biology tools to solve mathematical problems. Computing with DNA offers a completely new way of looking at and performing computations: the main idea is that data can be encoded in DNA strands, while molecular biology laboratory techniques (called bio- operations) that involve manipulation of DNA strands in test tubes can be used to simulate arithmetical and logical operations.

It was first introduced by Dr. Leonard Adleman of University of South California in 1994 who showed how to solve the Hamiltonian Path problem by manipulating the DNA strands in the tubes. Since then, more and more researchers are motivated by the promising future of this area and started working on it [17].

The basic idea of DNA computing arises from a mapping between the physics process in electronic computers and the chemistry process in DNA reactions. In electronic computers, everything is encoded in binary (0, 1) strings, while every DNA strand is encoded in four nucleotides[5]: A, T, G, and C. In electronic computers, the basic operations can be treated as manipulations on binary strings, while there are a bunch of biological operations on the DNA strands, e.g., concatenation (ligation), amplifying (copy), substitution, etc. These DNA operations can be performed in a controlled manner by modern biology technologies.

II. DNA BACKGROUND

DNA is a deoxyribonucleic acid , the genetic material that encodes the characteristics of living things. DNA consists of chemicals called nucleotides. There are four nucleotides in DNA[5] , each denoted by the first letter of its name : adenine (A), cytosine(C), guanine(G) and thymine(T). We can encode any information using this 4-letter alphabet , as we can encode any information in bits (0 and 1). It is now possible to synthesize strands of

DNA containing a specified sequence of nucleotides; that is , to create any desire string of letters to represent data.

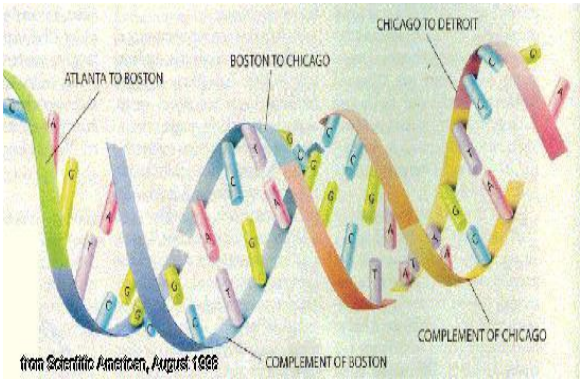


Figure 1[6] Double–stranded DNA , showing complementary pairs

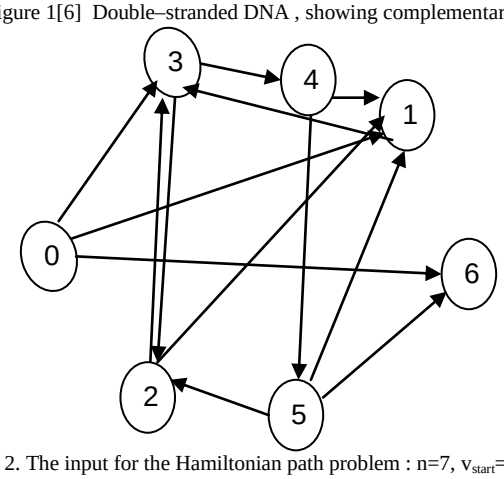


Figure 2. The input for the Hamiltonian path problem : n=7, $v_{start}=v_0$, $v_{end} = v_6$

John Watson and Francis Criok discovered double helix structure of DNA. The nucleotides form complementary pairs; A and T are complements, and C and G are complements. Two strands of nucleotides will attach to each other (and twist around each other in a double helix) if they have complementary elements in corresponding positions. For examples, Figure 1 (where we illustrate the attachment of complementary strands). The fact that complementary strands attach to each other is used repeatedly in the DNA algorithm for the Hamiltonian path problem. It can happen that two strands attach even though they do not have complementary elements in some position ; this is one of the properties of DNA processes that can cause problems for the algorithms.

2.1 Chemistry of DNA

DNA molecules : paired strands of nucleotides (bases A,T,C and G) attached to sugar phosphate backbones [7].

Backbone: 5 carbons (since carbons are present in sugar phosphate backbone) , similar to linked list[8].

- One molecule's 5' carbons binds to next one 3' carbons
- 1' binds to nucleotide

Paired Strands: In this we take molecules as

- Bases bond to complementary strand
- Sequences: listed 5' to 3'

Random pair of base is denoted by **S** is like **GCCATAGCTACGCCAT**

ISBN: 978-93-86171-35-1

which we called primers. These primers are used to represent the vertex of given graph. The complement of **S** is **s̲ CGGTATCGATGCGGTA**.

III. PCR(POLYMERASE CHAIN REACTION)

In this reaction we have a collection of DNA . This collection of DNA put between and two primers , we amplify strands of any sequence.

Given that: A collection of DNA and two primers, **s̲ , t**

Action : To amplify strands of the form **s̲vt** for any sequence of **v**.

Input: tube T of DNA , primers **s̲ , t**

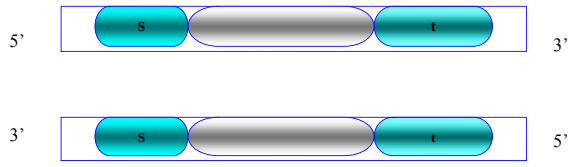
We repeat following step until satisfied.

- denature DNA with heat
- anneal DNA with primers
- elongate strands with DNA polymerase

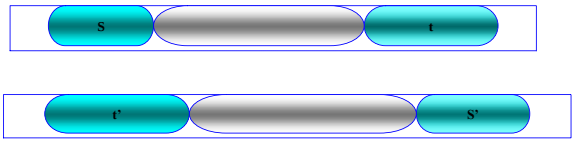
3.1. PCR examples

3.1.0 Given

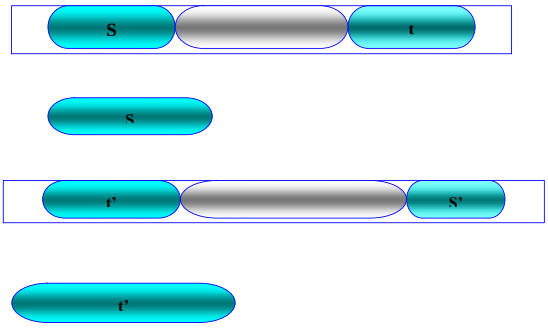
Here we take above given condition of 5' carbons binds to next one 3'carbons



3.1.1. Denature(heat)

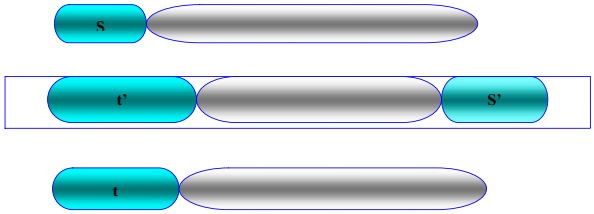


3.1.2. Anneal(Add primers)



3.1.3. Elongate(Add polymerase)

ISBN: 978-93-86171-35-1



IV. REPRESENTATION OF GRAPH IN DNA FORM(ADLEMAN'S ENCODING)

A vertex v in graph (V,E) is represented by the 20 random base pair sequence S_V

e.g S_2 **: TATCGCGATA GGTACCGGAT**

S_4: TGTGCTATGG GAACTCAGCG

S_6: AATGCTAGCT TAGCGATAGC

here subscripts 2 ,4 and 6 present the vertices name in graph (V,E). Edge (u,v) is represented by sequence S_{UV} which consists of last 10 base of S_U and first 10 base of S_V

e.g. **S_{24}: GGTACCGGAT TGTGCTATGG**

S_{46}: GAACTCAGCG AATGCTAGCT

Paths (u.v.w) is concatenations of edge(u,v) and edge(v,w)

e.g path(2.4.6)

GGTACCGGAT TGTGCTATGG GAACTCAGCG AATGCTAGCT

In fig 2. 0 is starting vertex and 6 is end vertex of graph(V,E) these vertices are also represented by S_0 and S_6 in DNA base form. Now we put all these sequence and path into tubes. We amplify tubes of S_V and $\underline{S_V}$ for each node v, amplify tubes of S_{UV} and $\underline{S_{UV}}$ for each edge (u,v). Then we mix all tubes into tube T with high probability , every path through G(V,E) will be represented in tube T. Now we do PCR operation on tube. Run PCR on tube T using S_0 and $\underline{S_6}$ as primers, we put products of tube T in tube T'. We separate strands with 20n+10 base from T' and put product of T' in R. If any DNA is left in R then return "YES" else "NO". Here 20n +10 indicate that number of base pair , n is natural number[9].

V. MATHEMATICAL MODEL FOR DNA FORM

We use some terminology of math which give convenient way to solve the problem[9]**.**

Detect(T): This function return true if there is any DNA in tube T otherwise return false.

Copy(T,{T_i}): This function pour T into each T_i .

Mix({T_i},T): This function pour all T_i into T.

Remove(T,T',{S_i}): Remove all String in T from S_i and placing them in T'.

VI. THE TRAVELING SALESPERSON PROBLEM

Input : For the traveling salesperson problem (TSP) , we are given a complete graph G=(V,E), v_i belongs to V, $1 \le i \le n$, where exists weight w_{ij} on edge v_iv_j

ISBN: 978-93-86171-35-1

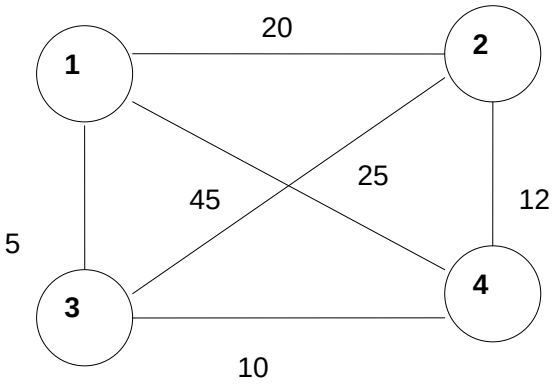


Fig 3. weight graph

Question : Is there any tour (a cycle through all the vertices can be visited exactly once) of minimum weight.

Output :

1.T_V contains S_V (and $\underline{S_V}$) , T_{UV} contains S_{UV} (and $\underline{S_{UV}}$)

// W_{ij} is weight (double type)

2. Mix({T_i , T'_{UV}},T)

3. while(Detect(T))

Copy(T ,{T_i})

If(T_i <> T_j and W_{ij} is minimum)

Then

VII. KNAPSACK AND SUBSET SUM PROBLEM

Input: Integer C , (s_1,s_2, . . . , s_n) and (p_1,p_2, . . . , p_n) a sequence of positive integers.

Question: The problem is to find a subset T of N (set of natural number) that maximizes the total profits, $\sum_{i \in T} p_i$ subject to constraint, $\sum_{i \in T} s_i \leq C$.

Output: Let we suppose that j is a natural number and T_j contains S_j (and $\underline{S_j}$). N contains all S_j (and $\underline{S_j}$) .

Here S_j (and $\underline{S_j}$) is 20 random pair sequence of DNA for j value, T_j ,T and N are tubes.

1. int maxSum=0, Sum, j

2. Mix({T_0},T)

3. For each T match with N

Sum= $\sum_{Tj \in T} s_j$

// s_j is integer value of sequence

If (Sum ≤ C) Then

For each T_j not in T

If (sum + s_j ≤ C) Then

Sum+=s_j

Mix({T,T_j},T)

Else

ISBN: 978-93-86171-35-1

```
            If (maxSum<Sum) Then
               maxSum=Sum
            End if
          End if
4.  If  (Detect(T)) Then
        Print maxSum
    Else
        Print "Error"
    End if
```

This algorithms take $O(n^2)$

VIII. CONJUNCTIVE NORMAL FORM(CNF) PROBLEM (MY APPROACH)

Input: Boolean formula in CNF with p-conjunction , q-literals per clause and n variables.

$F(x_n)= V^{p}_{i=1}(\cap^{q}_{j=1}\ l_{i,j})$

Where $l_{i,j}$ is either x_k or complement of x_k for some variable x_k .

Question: Is there an x_n such that $F(x_n)$ = True?

Output:

To solve this problem we use extra nodes which gives path and sequence of DNA form. This is called truth assignments. Here we have x1,x2,x3…xn as Boolean variable and we use extra v1,v2,v3…vn nodes. I suppose that x1', x2' x3'…xn' are the complements of Boolean variable x1,x2,x3 respectively.

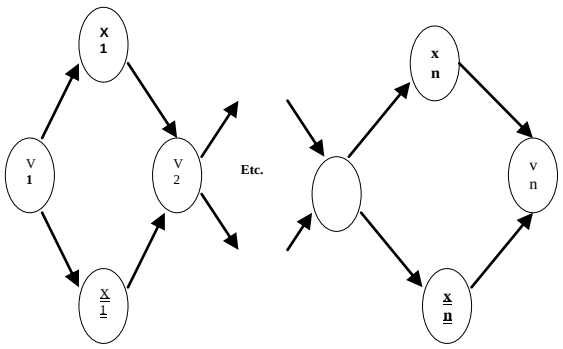


Fig: 4 Truth assignments graph.

1.We full all truth assignments to T_0.

```
2.  For i=1 to p
      For j=1 to q
       If l_{i,j} is positive Then
          Remove(T_{i,1},T',{x_j})                    Else
          Remove(T_{i-1} ,T',{x̄_j} )
      End if
```

ISBN: 978-93-86171-35-1

Re-label T' as T_i

4.If(Detect(T_p))

Then

return "YES"

Else

return "NO"

End if

This algorithms take $O(n^2)$

IX. BENEFITS OF DNA COMPUTING

9.1. Performs trillion of operation is simultaneously

We know that super computer can execute approximately 10^{12} operation per second[17] .Taking concatenation of DNA molecules (to generate paths) as a basic operation , it is estimated that the DNA method performed approximately 10^{12} operations per second and that number could be increased to about 10^{14}. At the higher rate, the number of operation per second would be more than 1000 times as many as executed by super computer[17]. The DNA method use less energy than a super computer. This comparison has to be interpreted carefully , however, because all the computer operation are directed by a program, while the DNA operations are only controlled , and are largely random.

9.2. Conduct large parallel calculations

Instructions in the electronic computers are much faster than the lab experiments (millions per second vs. one per hour or even day). But the DNA computer can potentially have more vast parallelism than the electronic computers. Therefore, the high parallelism will overcome the slowness of biological experiments. Furthermore, the lab experiments can be speed up once the manipulation of DNA strands can be done automatically by machines.

9.3. Energy efficiency

The energy cost of a DNA operation (on one strand) is about 1010 times less than the energy cost of an instruction in the electronic computers[10].

9.4. Economical data storage

One tube can store billions of DNA strands. One gram of DNA , which takes up about one cubic centimeter of space , can store as much information as one trillion compact disks.

X. MAIN OPEN ISSUES

10.1. Errors of the biological experiments. Usually, the error rate of biological experiments is much higher than the failure rate of electronic gates. The DNA strands are known to be fragile, they break easily. But along with the improvement of experimental technology and facilities, the error rate will be reduced.

10.2. How to make the DNA manipulation automatically? I believe this is just a matter of time.

10.3. Simple problems still require large amounts of memory

ISBN: 978-93-86171-35-1

XI. CONCLUSION

In this paper I tried to solve some well known NP-Complete problem by DNA computing .But the real implementation of DNA computer is more complicated as we think. It would be foolish to think that DNA computer can not be made. In future it must be designed completely and all algorithm, I proposed, will get right meaning as output in DNA computer.

DNA computing is a brand new research area which receives more and more attentions from both biologists and computer scientists. Some biological experiments has been performed which proved the possibility of DNA computing. Due to the highly parallel characteristics of DNA operations, the corresponding DNA algorithms scale well in the size of the problem. Therefore DNA computing shows potential advantages in solving the hard problems.

REFERENCES

[1] [Adleman, 1994] Adleman, L. M. Molecular computation of solutions to combinatorial problems. *Science*, 266(11):1021-1024, 1994.

[2] http://www.wikipedia.com/Hamiltonian path - Wikipedia, the free encyclopedia.htm

[3] http://www.wikipedia.com/travelling salesman problem - Wikipedia, the free encyclopedia.htm

[4] http://www.wikipedia.com/Conjunctive normal form - Wikipedia, the free encyclopedia.htm

[5] Franklin, Rosalind (1953). "Molecular Configuration in Sodium Thymonucleate. Franklin R. and Gosling R.G" (PDF). Nature 171: 740–741. doi:10.1038/171740a0. PMID 13054694.

[6] Form Scientific American: August 1998

[7] Bartlett & Stirling (2003)—A Short History of the Polymerase Chain Reaction. In: Methods Mol Biol. 226:3-6

[8] http://www.wikipedia.com/Polymerase chain reaction - Wikipedia, the free encyclopedia.htm

[9] http://people.ibest.uidaho.edu/~foster/Talks

[10] http://pages.cpsc.ucalgary.ca/~jacob/Courses/Winter2003/CPSC601-73/Slides

[11] [Boneh et al., 1995] Boneh, D., Dunworth, C., Lipton, R. J. and Sgall, J. On the computational power of DNA. *Technical Report TR-499-95*, Department of Computer Science, Princeton University, 1995.

[12] http://www.usc.edu/dept/molecular-science

[13] http://www.olympus.co.jp/en/magazine/techzone/vol54_e

[14] http://www.wikipedia.com/DNA computing - Wikipedia, the free encyclopedia.htm

[15] David I. Lewin (2002). "DNA Computing". Computing in Science & Engineering

[16] Aho, A.V., Hopcroft, J.E., and *Ullman, J.D.* (1976) The Design an d Analysis of Computer Algorithms, Reading, Mass., Addison-Wesley P.C.

Additional Literatures

[17] Sare Baase (San Diego State University) and Allen Van (University of California at Santa Cruz) Computer Algorithms “Introduction to Design and Analysis”

ISBN: 978-93-86171-35-1

[18] "Introduction to Algorithms" Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest , Clifford Stein .

[19] "Analysis and Complexity " P.P. Chakrabarti , IIT Kharagpur