

## Shortest Path Algorithm for DNA-computation

**Kamal Uddin Sarker**

Senior Lecturer  
Dept. of CSE (IT Academy)  
Northern University Bangladesh

Ku\_sarker@yahoo.com  
www.nub.ac.bd

example, which is performed in a constant time for 9-nodes and 16-edge graph. Actually it is a top-down design procedure. I will also show the complexity for this algorithm.

**Keywords:** PCR (Polymerase Chain Reaction), Gel Electrophoresis, Oligonucleotide.

### Abstract

**D**NA-computation is a latest inter-disciplinary issue and an efficient technique to solve no-polynomial problem by polynomial time. In this paper I will show a 5-step shortest path DNA-computation-algorithm with an

### Background

DNA computing known as a molecular computing is a new approach to massively parallel computation based on the groundbreaking work by Adleman [1]. He used DNA to solve 7-node Hamiltonian path problem, a special case of an NP-Complete problem that attempts to visit every node in a graph exactly once. In the last years there have been a lot of advantages in DNA computing [2]; DNA computers using dynamic programming enough to fit sequentially larger instances because of their large memory capacity than either conventional computers or previous brute force algorithms on DNA computers. The reason why dynamic programming algorithms are suitable for DNA computers is that the sub problems can be solved in parallel. A discovery in biology as the deoxyriboxyme-based (catalytic DNA) [3] produces an application in computing which is efficient for NP-Complete problems [7] and one of the latest ideas related to this field is the deoxyriboxyme-based logic gates [4] in a laboratory [8].

**DNA:** Deoxyribonucleic acid is a double-stranded sequence of four nucleotides (**Figure 1**): Adenine (A), Guanine (G), Cytosine (C) and Thymine (T), where the bonds are: A-T, T-A, C-G and G-C. Each DNA strand has two different ends that determine its polarity (3' and 5') where the double helix is anti-parallel (two strands of opposite polarity) bonding of two complementary strands. The internal structure (**Figure 2**) with different colors and two hydrogen bonds with A and T while three hydrogen bonds with C and G [5].

**Shortest path:** Let  $G=\{V,E\}$  be a directed graph with weight assigned to the edges, where  $V=\{v1,v2,\dots,vn\}$  is the set of vertices and  $E=\{e1,e2,\dots,en\}$  is the set of directed edges in the **graph-G**. Where weighted are mentioned at **Table 2** with correspondence edges. We have to find out the shortest path from source  $v1$  to destination  $v9$ .

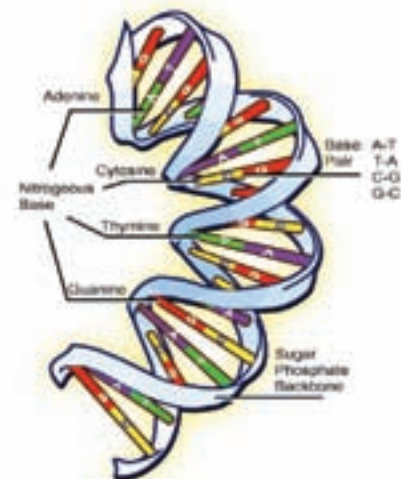


Figure 1

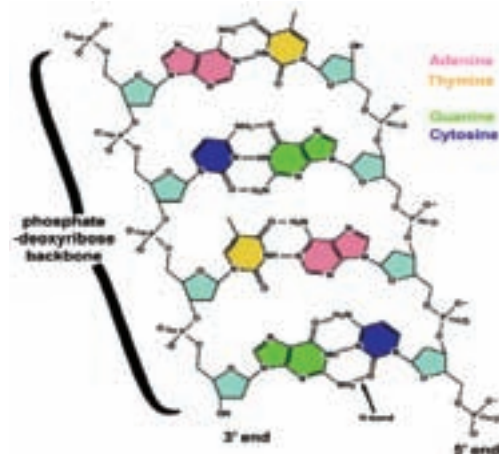
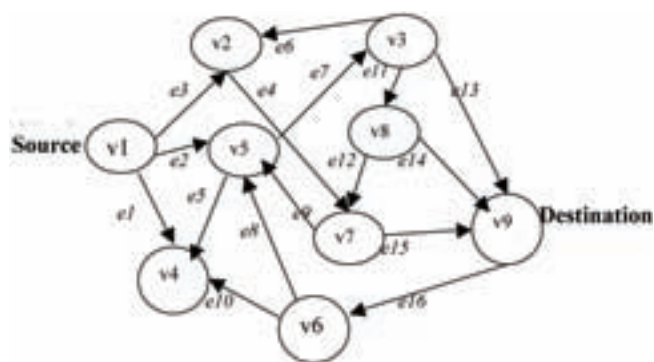


Figure 2



Graph-G

TABLE 1

| Vertex | DNA sequence   | Complement DNA sequence |
|--------|----------------|-------------------------|
| v1     | 5'ATCGATAGCT3' | 3'TAGCTATCGA5'          |
| v2     | 5'TAGCTATCGA3' | 3'ATCGATAGCT5'          |
| v3     | 5'CGTACGCATG3' | 3'GCATGCGTAC5'          |
| v4     | 5'GCTAGCGATC3' | 3'CGATCGCTAG5'          |
| v5     | 5'ATGCTTACGA3' | 3'TACGAATGCT5'          |
| v6     | 5'CCGTAGGCAT3' | 3'GGCATCCGTA5'          |
| v7     | 5'TACGAATGCT3' | 3'ATGCTTACGA5'          |
| v8     | 5'CGATAGCTAT3' | 3'GCTATCGATA5'          |
| v9     | 5'GCATCCGTAG3' | 3'CGTAGGCATC5'          |

TABLE 2

| Edge Decoding     | Weight | Edge Decoding      | Weight |
|-------------------|--------|--------------------|--------|
| e1-5'ATCGACGATC3' | 1      | e9-5'TACGATACGA3'  | 1      |
| e2-5'ATCGATACGA3' | 2      | e10-5'CCGTACGATC3' | -1     |
| e3-5'ATCGAATCGA3' | 3      | e11-5'CGTACGCTAT3' | 3      |
| e4-5'TAGCTATGCT3' | -1     | e12-5'CGATAATGCT3' | 2      |
| e5-5'ATGCTCGATC3' | 2      | e13-5'CGTACCGTAG3' | 1      |
| e6-5'CGTACATCGA3' | 1      | e14-5'CGATACGTAG3' | 2      |
| e7-5'ATGCTGCATG3' | -2     | e15-5'TACGACGTAG3' | 3      |
| e8-5'CCGTATACGA3' | 1      | e16-5'GCATCGGCAT3' | 1      |

MATRIX-M

| (J)    | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 |
|--------|----|----|----|----|----|----|----|----|----|
| (i) v1 | 0  | 1  | 0  | 1  | 1  | 0  | 0  | 0  | 0  |
| v2     | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |
| v3     | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |
| v4     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| v5     | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  |
| v6     | 0  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  |
| v7     | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  |
| v8     | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| v9     | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |

DNA Algorithm

Step-1: Encoding of the problem in DNAs.

An oligonucleotide is a short chain of nucleic acid, a set of nucleotides joined in a single strand. These chains are able to anneal with the complementary sequence of nucleotides, and a practical reason is that we are able to produce a specific oligonucleotide easily [6], which technique is used here. Each vertex in the graph has to be associated with a designed palindrome 10-mer sequence of DNA. For each edge  $e_i$  ( $v_i \rightarrow v_j$ ) in the graph are oligonucleotide 3'5-mer complementary sequence of  $v_i$  followed by 5'5-mer complementary sequence of  $v_j$  to be synthesized [1].

**Adjacent Matrix:** Suppose that  $G=\{V,E\}$  is a simple graph where  $|V|=n$ . Suppose that the vertices of G are arbitrarily as  $v1, v2, \dots, vn$ . The adjacency matrix **M** of **G**, with respect to this listing of the vertices, is the  $n \times n$  zero-one matrix with 1 as its  $(i,j)$ th entry when  $v_i$  and  $v_j$  are adjacent, and 0 as its  $(i,j)$ th entry when they are not adjacent. For the graph-G, adjacent matrix are given here **MATRIX-M**. It is used to identify edges which are decoding at **Table 2**.

Decoding Edges:

According to the rule mentioned above I have created the following edges. For  $e1$  ( $v1 \rightarrow v4$ ) last 5 complement characters from  $v1$  and first five complement characters from  $v4$  and weights are assigned.

Step-2: Construction of random path:

To construct random paths in the graph, a mixture containing each oligonucleotide encoding vertices and each oligonucleotide encoding edges has to be synthesized. Then take a pinch of each of the different sequences and put them into a test tube.

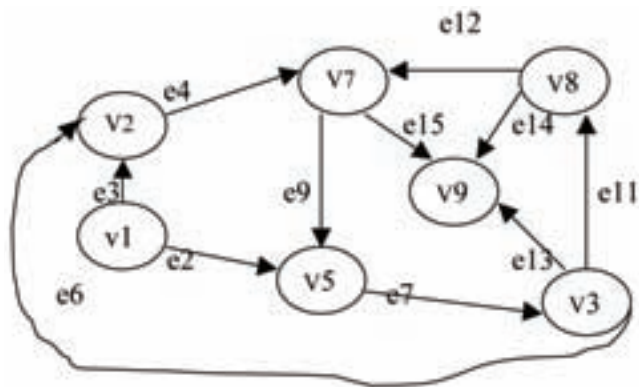
Construction of random paths:

- $e3 \rightarrow e4 \rightarrow e15$
- $e2 \rightarrow e7 \rightarrow e13$
- $e2 \rightarrow e7 \rightarrow e11 \rightarrow e14$
- $e2 \rightarrow e7 \rightarrow e11 \rightarrow e12 \rightarrow e15$
- $e3 \rightarrow e7 \rightarrow e15$
- $e2 \rightarrow e7 \rightarrow e6 \rightarrow e4 \rightarrow e15$
- .....

Step-3: Amplification of DNA paths by PCR (Polymerase Chain Reaction).

Amplification of DNA paths that begin with a vertex source and end with a vertex destination is to be performed. Two specific primers that can anneal with source vertex and destination vertex are to be added to the PCR reaction. In our example, if we take the source as vertex-1 and the destination

as vertex-9, the primer corresponding to the source and destination (TAGCT (first part complement of v1) and CGTAG (last part of v9) respectively) are added to the PCR, we obtain the following paths, and simplifying graph-g. We can sort according to the length of the path by Gel Electrophoresis [8].



Graph-g

**Step-4: Elimination of repetitions**

There are two loops existing which used to repetitions of  $v7 \rightarrow v5 \rightarrow v3 \rightarrow v8 \rightarrow v7$  and  $v2 \rightarrow v7 \rightarrow v5 \rightarrow v3 \rightarrow v2$  are removed [9] by looking at the formation of Haippin looped structure. That means the path having the repetition of nodes is not considered in deriving the solution. After removing,

Path according to vertices:  $v1 \rightarrow v5 \rightarrow v3 \rightarrow v9$  (5'ATCGA TAGCTATGCTTACGACGTACGCATGGCATCCGTAG3')  
 Path according to edges:  $e2 \rightarrow e7 \rightarrow e13$  (3'ATCGATACGA ATGCTGCATG CGTACCGTAG5')  
 Total weight for this path:  $2-2+1=1$

Path according to vertices:  $v1 \rightarrow v5 \rightarrow v3 \rightarrow v8 \rightarrow v9$  (5'ATC GATAGCTATGCTTACGACGTACGCATGCGATAGCT ATGCATCCGTAG 3')  
 Path according to edges:  $e2 \rightarrow e7 \rightarrow e11 \rightarrow e14$  (3'ATCGAT ACGAATGCTGCATGCGTACGCTATCGATACGTAG5')  
 Total weight for this path:  $2-2+3+2=5$

Path according to vertices:  $v1 \rightarrow v5 \rightarrow v3 \rightarrow v8 \rightarrow v7 \rightarrow v9$  (5'ATCGATAGCTATGCTTACGACGTACGCATGCGA TAGCTATTACGAATGCTGCATCCGTAG3')  
 Path according to edges:  $e2 \rightarrow e7 \rightarrow e11 \rightarrow e12 \rightarrow e15$  (3'ATCGATACGA ATGCTGCATG CGTACGCTAT CGATAATGCT TACGACGTAG5')  
 Total weight for this path:  $2-2+3+2+3=8$

Path according to vertices:  $v1 \rightarrow v2 \rightarrow v7 \rightarrow v9$  (5'ATCGATAGCT TAGCTATCGA TACGAATGCT GCATCCGTAG 3')  
 Path according to edges:  $e3 \rightarrow e4 \rightarrow e15$  (3'ATCGAATCGA TAGCTATGCT TACGACGTAG5')  
 Total weight for this path:  $3-1+3=5$

Path according to vertices:  $v1 \rightarrow v2 \rightarrow v7 \rightarrow v5 \rightarrow v3 \rightarrow v9$  (5'ATCGATAGCT TAGCTATCGA TACGAATGCT ATGCTTACGA CGTACGCATG GCATCCGTAG3')

Path according to edges:  $e3 \rightarrow e4 \rightarrow e9 \rightarrow e7 \rightarrow e13$  (3'ATCGAATCGA TAGCTATGCT TACGATACGA ATGCTGCATG CGTACCGTAG5')

Total weight for this path:  $3-1+1-2+1=2$

Path according to vertices:  $v1 \rightarrow v2 \rightarrow v7 \rightarrow v5 \rightarrow v3 \rightarrow v8 \rightarrow v9$  (5'ATCGATAGCT TAGCTATCGA TACGAATGCT ATGCTTACGA CGTACGCATG CGATAGCTAT GCATCCGTAG3')

Path according to edges:  $e3 \rightarrow e4 \rightarrow e9 \rightarrow e7 \rightarrow e11 \rightarrow e14$  (3'ATCGAATCGA TAGCTATGCT TACGATACGA ATGCTGCATG CGTACGCTAT CGATACGTAG5')

Total weight for this path:  $3-1+1-2+3+2=6$

Path according to vertices:  $v1 \rightarrow v2 \rightarrow v7 \rightarrow v5 \rightarrow v3 \rightarrow v8 \rightarrow v7 \rightarrow v9$  (5'ATCGATAGCT TAGCTATCGA TACGAATGCT ATGCTTACGA CGTACGCATG CGATAGCTAT TACGAATGCT GCATCCGTAG3')

Path according to edges:  $e3 \rightarrow e4 \rightarrow e9 \rightarrow e7 \rightarrow e11 \rightarrow e12 \rightarrow e15$  (3'ATCGAATCGA TAGCTATGCT TACGATACGA ATGCTGCATG CGTACGCTAT CGATAATGCT TACGACGTAG)

Total weight for this path:  $3-1+1-2+3+2+3=9$

**Step-5: The final path obtained**

After finding the weights of DNA strands, we can easily observe that the path

$v1 \rightarrow v5 \rightarrow v3 \rightarrow v9 = e2 \rightarrow e7 \rightarrow e13 = 2-2+1=1$  is minimum weight/cost and whose correspondence DNA sequence is given by ATCGATACGAATGCTGCATG CGTACCGTAG (edges sequence).

**Complexity of DNA algorithm**

1. Annealing (construct random paths) is performed within a constant time in a microenvironment (test tube) maintaining desired temperature required for the reaction.
2. A constant time is also needed to identify valid paths from random paths by PRC, just addition primers and wait for reaction.
3. Within a constant time the repetition vertices or edges are removed by SSCP.
4. Sequencing is automated and hence its need for a fixed time to automatically provide the sequencer (ABI Prism 3700) in a printed format which is called the chromatogram so it obtains the desire solution is also constant.

Thus all the reactions can be performed in a constant time irrespective of the number of nodes and edges in the graph, and hence the proposed DNA algorithm is of polynomial time.

**Computer's Algorithm:**

**Input:**

N-Number of nodes in the graph  
 $V_i$ -10-base distinct palindrome DNA sequences  
 $L(i, j)$ -Adjacence Metrix  
 $W_i$ -Weights of the edges  
 $V_s$ -Source  
 $V_d$ -Destination  
 $K \leftarrow$  all possible paths from source to destination  
 $P \leftarrow$  A path from source to destination  
 $PMW \leftarrow$  Path from source to destination for minimum weight

**Step-1:**  $N \leftarrow$  number of nodes

**Step-2:** For  $i=1$  to  $N$

$V_i \leftarrow$  Assign palindrome DNA sequences (ATCGTAGC)

**Step-3:** For  $i=1$  to  $N$

For  $j=1$  to  $N$

$L(i, j) \leftarrow$  Adjacency Matrix

If  $L(i, j) = 1$  then  $E_{ij} \leftarrow$  Edge Sequence

**Step-4:** For  $i=1$  to  $N$

For  $j=1$  to  $N$

Generate-Path ( $i, j$ ) creates all path by back tracking  
 Run-Primers( $V_s, V_d$ ) gives all paths which starts from source and ends to destination.

**Step-5:** For  $i=1$  to  $k$

$W_i = \text{Read}(P_i)$

$PMW \leftarrow$  Minimum ( $W_i$ )

**Complexity of Computer's simulation**

For step-2, the validity of DNA strands, in the sense that, whether they are palindrome or not. So this is for order  $N^2$ . In step-2 two loops are running of  $N$  elements so there is also  $N^2$ .  $N^2(N-2)^e$  for all path in graph by backtracking where  $N$  is the number of vertices and  $e$  is the edges.

Complexity:  $N^2 + N^2 + N^2(N-2)^e \sim o(N^2(N-2)^e)$

**Conclusion**

Though it is a theoretical solution, any-one (computer scientist, biologist or chemist) can apply it if he/she has the laboratory facilities. In the same way, we can solve Euler path, graph coloring or real time problem in mapping section. In addition I would like to mention that I have been working to design combination logic gates.

**References**

[1] Leonard M. Aldeman "Molecular computation of solutions to combinatorial problems", Science 266, 1021-1024, 1994.  
 [2] David I. Lewin "DNA computing", IEEE, 2002.  
 [3] R.R. Breaker, G.F. Joyce, Chemical Biology 2, 655-660, 1995.  
 [4] Milan N. Stojanovic, Tiffany Elizabeth Mitchell, and Darko Stefanovic "deoxy-riboxyme-based logic gates", American Chemical Society, 2001.  
 [5] J.D.Waston & F.H.C. Crick "A structure for DNA", Nature, 1953.  
 [6] M.B.C. Ruiz-Perez and Dr. A. Virazel " Innovative computer architectures and concepts seminar, July-2002, universität Stuttgart.  
 [7] Lipton RJ, "DNA solution of hard computational problems", science, 268:542-545, 1995.  
 [8] M. Amos, G. Paun, Grzegorz, Rozenberg and Arto Salomaa, Topics in the theory of DNA computing," Theoretical Computer Science, 287, 2000, 3-38.  
 [9] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyoma, T. Yokomori and M. Hagia, "Molecular computation by Hairpin formation," Science, 288, 2000, 1223-1226.