

## Solving Shortest Hamiltonian Path Problem Using DNA Computing

Hala Mohammed Alshamlan  
Computer Science Department  
King Saud University  
Riyadh, Saudi Arabia  
[halshamlan@ksu.edu.sa](mailto:halshamlan@ksu.edu.sa)

Mohammed El Bachir Menai  
Computer Science Department  
King Saud University  
Riyadh, Saudi Arabia  
[menai@ksu.edu.sa](mailto:menai@ksu.edu.sa)

**Abstract—** Deoxyribonucleic acid (DNA) computing fundamentally being similar to parallel computing provides a nice way to make trillions of similar calculations in less than moment. Moreover, DNA computing has ability to solve main NP-complete problems such as Hamilton Path Problem, 3-SAT Problem, and Maximum Clique Problem. In this paper, we apply DNA computing to solve Shortest Hamiltonian Path Problem using two steps. First one, determine all Hamiltonian path from specific weighted graph, and then, in second step, we select the shortest one and return it as solution of our problem.

**Keywords—** Shortest Path Problem; Hamiltonian Shortest Path Problem; DNA Computing; NP Hard Problems.

### I. INTRODUCTION

DNA computing is a form of computing which uses DNA, biochemistry, and molecular biology, instead of the traditional silicon-based computer technologies [10]. DNA computing, or more generally molecular computing, is based on manipulations with DNA strands using some basic biological transformations. Being very similar to parallel computing, DNA computing promises to solve many NP-complete problems, much faster than modern silicon-based computers do [13].

Leonard Adleman of the University of Southern California initially developed this field. In 1994, Adleman demonstrated a proof-of-concept use of DNA as a form of computation, which solved the seven-point Hamiltonian path problem (HPP) [2]. HPP is to find an air flight path from given cities such that each city is visited once and only once. Therefore, HPP is NP-complete Problem [1][2][9][12]. Particularly, since Adleman solved a small instance of the Hamilton path problem successfully, the DNA computing has become a new focus in the scientific areas of nanotechnology, biology, mathematics, medicine and information science [4][5]. Compared with electronic computers, which often need exponential time, DNA computing has its own advantages. With its huge parallelism for computing, almost  $10^{18}$  information data can be proceeded in parallel [2].

After that, a major goal of subsequent research is how to use DNA manipulations to solve P and NP-hard problems, especially 3-SAT problems [17]. 3-SAT is one of NP-complete problems, and it is as hard as all the other NP problem, which is to search for a model (or solution) of a set of clauses with each clause composed of no more than three

literals, where a literal is a variable (or an atom) or its negation. Various solutions were tried to solve the 3-SAT problem, Lipton [6] proposed DNA experiments on test tubes to solve a satisfiability problem based on DNA sticker computing model. Later, Ouyang used short linear dsDNA molecules and DNA restriction enzymes to solve maximal clique problem [7]. After that, another way for DNA computing was developed, In 2000, Liu et al. [8] introduced a new simple case and method to solve a 3-SAT problem, in which the feasibility of DNA surface computing was verified and also proved that the fluorescence could be used accurately in DNA computing.

Moreover, there are many weight encoding method have been studied in literature like in [14][15][16][17]. Thus, further study of the DNA encoding of weight is very important. Notably, the weight encoding methods are mostly used to Finding simple shortest path problem, which is P problem. Based on the existing literature, we show many methods and algorithms proposed to solve shortest path problem [14] [15][17], but these methods can only solve a specific examples and there are some limitations and concentrations in these methods. In other hand, Hu et al. in [16] proposed an effective and new IMCE encoding method based on Incomplete Molecule Commixed Encoding and use this method to find the shortest path of a seven vertex weighted graph. In order the shortest path problem is a variant of the Hamiltonian path problem in that it asks for the shortest route/path between two given nodes, and because the methods proposed in [16] is very effective comparing to other proposed methods in literature, we apply the Incomplete Molecule Commixed Encoding (IMCE) encoding method proposed in [16] in weighted directed graph to investigate the solution of the shortest Hamilton path problem using DNA computing.

The rest of this paper organized as follow: Since the DNA computing required understanding of biological structure and operation of DNA, we will present general background about this issue in Section 1. In Section 2, we define our problem in formal way. After that, in Section 3, we will illustrate how we use DNA stand to encode shortest Hamiltonian Path Problem. In Section 4, proposed DNA algorithm will be presented. Then, our result and analysis will be demonstrated in Section 5. Finally, our conclusion and contribution appear in last section.

## II. DNA BIOLOGICAL BACKGROUND

In order to understand the DNA computing application, and because the DNA molecular is little bit consider as sophisticated for computer scientist reader, we give the reader foundation about the molecular biology. In this section, we present the DNA biological structure and the main DNA operation.

### A. DNA Structure

The (deoxyribonucleic acid) DNA stand is encodes the genetic information of cellular organisms. It consists of polymer chains, commonly referred to as DNA strands [1]. Each strand may be viewed as a chain of nucleotides, or bases, attached to asugar-phosphate “backbone”. An n-letter sequence of consecutive bases is known as an n-mer or an oligonucleotide (commonly abbreviated to “oligo”) of length n. Strand lengths are measured in base pairs (b.p.). These complex molecules are composed of basic blocks called nucleotides, nucleic acid bases A (adenine), G (guanine), C (cytosine), and T (thymine) [or U (uracil) in RNA] [5]. Each strand, according to chemical convention, has a 5’ and a 3’ end, thus, any single strand has a natural orientation. The two pairs of bases form hydrogen bonds between each other, two bonds between A and T, and three between G and C. Each base has a bonding surface, and the bonding surface of A is complementary to that of T, and that of G is complementary to that of C. This complementary rule is called Watson–Crick complementary. A single DNA strand can pair with another strand when their sequences of bases are mutually complementary and the chains have opposite polarity [1]. Here is an example of a double stranded DNA chain.

```
5'  CCCAATGAACCCCATTT  3'
3'  GGGTACTTGGGGTAAA  5'
```

### B. DNA Operations

Some (but not all) DNA-based computations apply a specific sequence of biological operations to a set of strands. These operations are all commonly used by molecular biologists. In this section we describe the basic and more important of them in more detail.

#### 1) Synthesis

The synthesizer is supplied with the four nucleotide bases in solution to obtain randomly all possible solutions, which are combined according to a sequence entered by the user. The instrument makes millions of copies of the required oligo and places them in solution in a small vial [18][20].

#### 2) Denaturing, annealing, and ligation

Double-stranded DNA may be dissolved into single strands (or denatured) by heating the solution to a temperature determined by the composition of the strand [2]. If a molecule of DNA in solution meets its Watson-Crick complement, then the two strands will anneal that is, twist around each other to form the famous double helix. In other hand, annealing is the reverse of melting, whereby a solution of single strands is cooled, allowing complementary strands to bind together.

In double-stranded DNA, if one of the single strands contains a discontinuity (i.e., one nucleotide is not bonded to its neighbor), then this may be repaired by DNA ligase. This particular enzyme is useful for DNA computing, as it allows use to create a unified strand from several strands bound together by their respective complements. [9][18]. DNA ligase is used by the cell to repair breaks in DNA strands.

#### 3) Separation of strands

This operation mainly use when we need select specific solution or in filtering step. For this, we may use a “molecular sieving” process known as **affinity purification**. If we want to extract from a solution single strands containing the sequence x, we may first create many copies of its complement, x. We attach to these oligos biotin molecules (a process known as “biotinylation”), which, in turn, bind to a fixed matrix [18].

#### 4) Gel electrophoresis

The contents of a test tube can be separated by increasing length. This is achieved by gel electrophoresis, whereby longer strands travel more slowly through the gel. Electrophoresis is the movement of charged molecules in an electric field [20]. Since DNA molecules carry a negative charge, when placed in an electric field, they tend to migrate toward the positive pole. The negatively charged DNA molecules move toward the anode, with shorter strands moving more quickly than longer ones [4]. Hence, this process separates DNA by length

#### 5) PCR

PCR employs polymerase to make copies of a specific region (or target sequence) of DNA that lies between two known sequences. In order to amplify template DNA with known regions (perhaps at either end of the strands), we first design forward and backward primers (i.e. primers that go from 5’ to 3’ on each strand. We then add a large excess (relative to the amount of DNA being replicated) of primer to the solution and heat it to denature the double-stranded template. Cooling the solution then allows the primers to anneal to their target sequences. We then add the polymerase, which extends the primers, forming an identical copy of the template DNA. Thus, if we then repeat the cycle of heating, annealing, and polymerising, it is clear that this approach yields an exponential number of copies of the template (since the number of strands doubles after each cycle) [18][20].

## III. PROBLEM DEFINITION

In this research paper, our main problem is how we identify shortest Hamiltonian path from specific weighted graph using Incomplete Molecule Commixed Encoding (IMCE) model. Before we describe our proposed algorithm that is based on IMCE model to solve the shortest Hamiltonian path problems, we give some description about incomplete molecule. In IMCE computing model or in other word incomplete molecules used in this scheme like the domino shown in Figure 1. The biological operations of these molecules are similar with sticker model.

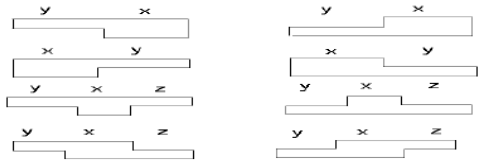


Figure 1. The molecular structure of domino

Sticker model is a universal computing system, which is used by Adelman [1][9] and Lipton [6]. In DNA biological operation, Restriction Enzymes or Nucleases are used to cut and stick the strands. The principle of this model is shown as Figure 2, the logic of the sticker model presented in [21], which is based on the paradigm of Watson–Crick complementarity. In short, the model involves a long single memory strand and a number of sticker strands or stickers as indicated in Figure 2. A memory strand is a single-stranded DNA with  $n$  bases. It is divided into  $k$  non-overlapping substrands, each of which has  $m$  bases, and therefore,  $n = km$ . Each sticker has  $m$  bases and complementary to exactly one of the  $k$  substrands in the memory strand. During a course of computation, each substrand is identified as a Boolean variable and is considered “true” or “false” as its corresponding sticker is annealed or not.

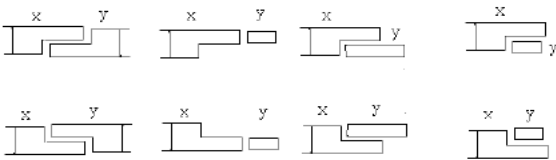


Figure 2. The structure of Sticker model.

Later on, in this paper, we will describe in detail how the IMCE Encoding Scheme solves the shortest Hamiltonian path problems. Therefore, we will define the meaning of abbreviations used in IMCE scheme as follows:

- VE( $V_i$ ): vertex encoding of  $V_i$ .
- WE( $W_i$ ): weight encoding of  $W_i$ .
- EE: edge encoding

To solve our problem, first, we will determine all Hamiltonian paths from specific weighted graph  $G$ , and then we will select the shortest one and return it as solution of this problem. For clarification, given a weighted graph  $G=(V, E)$ , the vertex set is  $V$ , the weight set is  $W$ , the Edge set is  $E$ , where  $w \in W$  (see Figure 3).

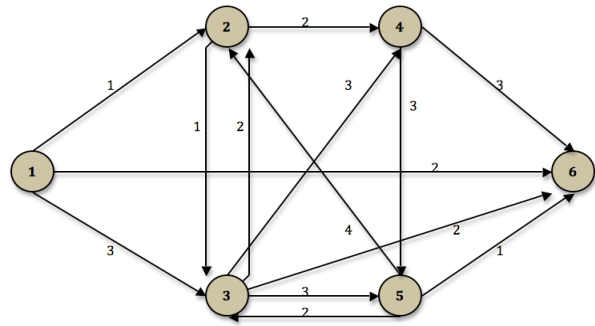


Figure 3 . A weighted Graph G

In Graph  $G$ , there are many Hamiltonian paths presented in Table 1. Each one of them has specific weight, and based on these weights we can determine that the last path is the shortest Hamiltonian path in graph  $G$ , which is has the weight equal 9.

TABLE 1. IDENTIFY SHORTEST HAMILTONIAN PATH FROM WEIGHTED GRAPH G

Hamilton Path	Weight
$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 6$	$1+2+3+2+2=10$
$1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$	$3+2+2+3+1=11$
$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$	$1+1+3+3+1 = 9$ <b>(The Shortest Hamilton Path)</b>

Adelman [1] applied five steps to solve Hamiltonian path problem (HPP) using DNA computing. Our problem in this study is similar to HPP with additional two steps and update one-step. The first new step is weight encoding and representation. Notably, we consider this step with vertexes and edges encoding as first step, and we must do it before the path construction. Because the weight of each edge is varying, in this problem we cannot determine exactly which paths was visited exactly  $n$  vertexes like HPP. For this reason, we update step four to become more suitable in this problem. In addition, the second new step is to select the shortest Hamiltonian path.

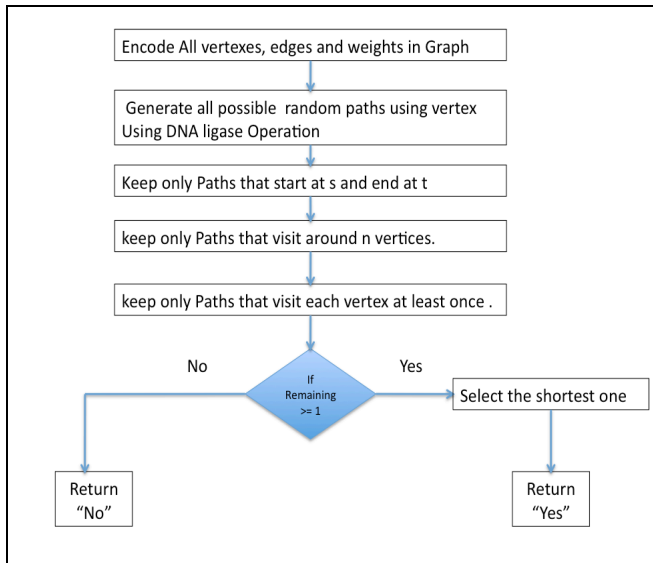


Figure 4. our proposed methodology

Moreover, in this study, we proposed seven main steps that demonstrate our methodological description to solve shortest Hamilton path problem (see Figure 4), as follows:

1. Encode the vertexes, edges and weights.
2. From step1, generate random paths using vertex
3. From all paths created in step 2, keep only those that start at s and end at t.
4. From all remaining paths, keep only those that visit around n vertexes.
5. From all remaining paths, keep only those that visit each vertex at least once.
6. If the path remaining more than 1 then From all remaining paths, keep only the shortest path.
7. If any path remains return, "yes" with the path ;otherwise, return "no".

#### IV. DNA ENCODING

In this section, we illustrate in detail how we can encode and representing vertexes, weights, and edges in direct graph. In our problem the vertex encoding is similar to Hamiltonian path problem presented in [1][9]. Therefore, we need to investigate specific encoding technique to encoding and representing the weight and then representing the edges.

##### A. Vertex Encoding

Basically, we adopt the length of single strand is 20 mer, because this length is widely applied by many researcher in DNA computing field like by Adelman [1][9] and Lipton [6]. Each encoding of the vertex is unique. For the instance shown in Figure 3, the encoding of vertex V1, V2 and V6 are as follows:

VE(V1)= ATGCAAGGTC TGACGGTTCA  
 VE(V2)= GATCGGTAAC GACTGGTTAC  
 VE(V6)= TACGGTTACA TCGATTGAAA

##### B. Weight Encoding

There are important definitions that must be illustrated before we introducing the method of weight encoding, we give as follows:

###### • Definition 1

Weight Set: The set consisted with the weights of the edge in a weighted graph G. For instance, the Weight Set of the graph shown in Figure 3, each edge has different weight; some of them are 1, 2, or 3. Thus, the weight set of graph G is  $A=\{1,2,3\}$ .

###### • Definition 2

Minimum function (MIN): Return the minimum value in the Weight Set. For instance,  $MIN (A)=1$ .

###### • Definition 3

Complementary strand Mapping function (H): Following the principle of Watson-Crick Complementary, this function will return a complementary base sequence of a sequence. For instance:  $H (ATTGCA)=TAACGT$ .

###### • Definition 4

Distribution Ratio: It indicates the disparity between the weights and their average of a given graph, all the elements of the average difference of A is the resolution of graph. Its formula is:

$$\frac{\sum_{i=1}^n \sum_{j=i+1}^n (W_i - W_j)}{C_n^2 + n}$$

Where  $W_i$  and  $W_j$  are the weights of edge i and edge j, and they belong to weight set A. Where n is the number of elements of edges in A. For instance, the Distribution Ratio of the graph shown in Figure 3 can be calculated as:

$$\begin{aligned} &= \frac{(2-1) + (3-1) + (3-2)}{C_3^2 + 3} \\ &= \frac{(2-1) + (3-1) + (3-2)}{1+3} \\ &= \frac{(1) + (2) + (1)}{4} \\ &= \frac{4}{4} = 1 \end{aligned}$$

Now, we begin to illustrate the encoding method of weight. The weight is encoded as double-strand with variable-length. As mentioned previously, in DNA Biological structure, A and T pair form two hydrogen bonds while G and C form three hydrogen bonds. By use of this difference, we use G and C pair to express 1, while A and T pair to express 0. Using this definition, we can translate the weight to binary string. Meanwhile, The binary encoding of 1 = "01" and the binary encoding for 0 = "00". We consider the length of the binary string is variable. Now, the question is: can we use the length of DNA strand in this encoding to distinguish the different length of path of a weighted graph? Let us use an example to illustrate. Supposing from one

vertex to another vertex can be directly reached by an edge which weight is 5. It can also be arrived by two edges with weights of 2 and 2. Obviously,  $5 > 2+2$ . How about the length of the binary string mentioned above? The binary encoding of 5 is "101". The binary encoding of 2 is "10". So, the string length of 4 is "1010". (Here, do not consider the encoding length of the vertex). Obviously, the length of "1010" is larger than "101". But  $4 < 5$ , so this method is invalid. We cannot use the length of encoding to judge the problem of the shortest Hamiltonian path. In order to solve this problem, we introduce the concept of Distribution Ratio. For the given graph, as shown in Figure 3, the weight set  $A = \{1, 2, 3\}$ , and  $\text{MIN}(A) = 1$  by the definition above, the Distribution Ratio of A is 1, we can encode based on the minimum value of MIN and Distribution.

So, in our example, the binary encoding of 1 = "01". Thus, the DNA molecules can be expressed as follows:

C T  
WE (1) = G A  
2 = 1 + 1. The binary encoding is "0101",  
G A G A  
WE (2) = C T C T  
3 = 1 + 1 + 1. The binary encoding is "010101",  
C T G A G A  
WE (3) = G A C T C T

### C. Edge Encoding

Edge encoding depends on the above vertex encoding and weights encoding methods. We applied incomplete molecular form in our proposed edge encoding method. To represent the edge, first, we put two single-strand (ss) represent the vertex encoding named ( $V_i$  and  $V_j$ ) into the test tube. Each single strand divided on two half, for instance  $V_i$  consist of  $V_i'$  as first half, and  $V_i''$  as second half. After that, we put the incomplete double strand (ds) in the same tube, which is represent the weight encoding of edge between vertexes  $V_i$  and  $V_j$  named  $W_{ij}$ . Figure 5 illustrates the structure of an edge.

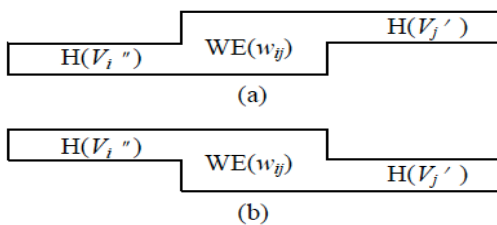


Figure 5. The structure of edge encoding

In Figure 4, the structure of incomplete molecule consists on three parts. The first part  $H(V_i'')$  is the complementary sequence of  $V_i'$ , which is the second half of the single strand of vertex  $V_i$ . The second part is  $WE(w_{ij})$ , which is double-strand represent the edge weight between vertex  $V_i$  and  $V_j$ . And the third part  $H(V_j')$  is the complementary sequence of  $V_j'$ , which is the first half of the single-strand of vertex  $V_j$ . Moreover, Each edge in the graph shown in Figure 3 should be encoded to two incomplete double-strands as described in

Figure 4. For example, the encoding of the edge from vertex 1 to vertex 2 with weight 1 in Figure 3 has two incomplete double strands expressed as follow:

$EE(V_1-V_2) = H(V_1'') + WE(1) + H(V_2')$   
CTAACTAGCCATTG  
ACTGCCAAGTGA  
  
And  
ACTGCCAAGTCT  
GAAACTAGCCATTG

Following the above method, we can encode any vertex in graph shown in Figure 3. Notably, that  $V_i$  is equal ( $V_i' + V_i''$ ). And, we express it by 20bp oligonucleotide fragments, where the length of  $V_i'$  and  $V_i''$  are both 10bp (All the direction of encoding is  $5' \rightarrow 3'$ ). Then, we can calculate the weight set of the graph and the distribution ratio. After that, we start encoding the weights based on the above algorithm, and then encode the incomplete molecule structure of each edge.

### V. DNA ALGORITHM

In this section, we will explain each step in our proposed DNA algorithm to solve shortest Hamiltonian path problem in more details.

#### Step 1: Encode the vertexes, edges and weights

We already illustrated this step in previous section.

#### Step 2: Generate Random paths.

In this step, we will mix first tube contain the vertex encoding and the second tube that contain the edge encoding with weight into one tube T. Then in T many ligase operation reactions will take place.

#### Step 3: keep only those that start at s and end at t.

In this step, we use the PCR operation using the prime of first and end vertexes. Suppose the first vertex is  $V_2$  and the end vertex is  $V_6$ . Thus, in this step we enlarge the reaction for vertex  $V_2$  and  $V_6$  and the number of strands that begin at  $V_2$  and end at  $V_6$  will sharply increase. But, the number of other strands does no change.

#### Step 4: keep only those that visit exactly n vertexes.

The result of this step is approximated, because in this problem we already express the vertex encoding by 20bp oligonucleotide fragments. But, we cannot determine the weight for each edge in the path. Suppose N is the number of vertexes in graph, MDR is the minimum value of MIN and Distribution Ratio, and AVG is the average of weight set A.

In this step, we use the Gel Electrophoresis, and the length L of accepted stand should be as follows:

$$((N+1) * 20) + ((N+1) * |AVG|) < L < (N * 20) + (N * |MDR|)$$

**Step 5: keep only those that visit each vertex at least once.**

In this step, we make strands separation by using sequence of Affinity Purification operation using the compliment of each vertexes in graph many times until reach that each vertex appears exactly one time in each path in tube.

**Step 6: keep only the shortlist path.**

DNA strands generated from step 5 can be separated in terms of its length by means of gel electrophoresis. The molecules are separated according to their weight, which is almost proportional to their length. Because each edge (i,j) has three parts ( H(Vi'') + WE (Wij) + H(Vj') ), and H(Vi'') , H(Vj') has equal length which is 10 bp ,we concludes that the different between edges in WE(Wij) part. Therefore, the longest path has long molecule stand.

**Step 7:Obtaining the Answer.**

In the last step, we need to use PCR prime operation and Nucleases (primers) operation to determine the order of each vertexes in shortest Hamiltonian path problem as follow:

Conduct a “graduated PCR” using a series of PCR amplifications. Use primers for the start vertex s and the n<sup>th</sup> item in the path. So, to find where vertex x lies in the path you would conduct a PCR using the primers from vertex s to vertex x, and by using the following proposed algorithm:

// Suppose that L is the length form vertex s to vertex x

1.  $\bar{L} = \text{Round}(L/20)$ .
2.  $E = (\bar{L} - 1) * |AVG|$   
Where E is Expected weight, and AVG is Average of weight set A.
3. Subtract the expected weight form the length,  $L = L - E$ .
4. The order of vertex x = round (L / 20).

VII. ANALYSIS AND RESULT

In this section, we proved the efficiency of our proposed algorithm, by present two practical examples.

**First example:** Suppose that we have graph G that contains two shortest Hamiltonian paths, the first on weight equal 5 and the second one weight 3. The first path weight represented as follows:

GTGTGTGTGT  
CACACACACA

So, in this path we required 10 mer. On other hand, the second path weight represented as follows:

GTGTGT  
CACACA

Here, in the second path, we required only 6 mer. We observed that the first path is longest than the second path. Therefore, when we put them in the gel electrophoresis we will observe that the second path move faster. However, this technique is based on the fact that DNA molecules are negatively charged. Hence, by putting them in an electric

field, they will move towards the positive electrode at different speeds. The longer molecules will remain behind the shorter ones.

**Second example:** Suppose the length from s the vertex x is 64. So, L = 64 base pairs. (i.e. 4 base for the weight : 1+1= 0101). To determine where vertex x lies in this path, we adopt our proposed algorithm as follow:

1.  $\bar{L} = \text{Round}(64/20)$ ,  $\bar{L} = 3$
2.  $E = ((\bar{L} - 1) * 4)$ , (i.e.  $4 = |AVG| = |0101|$ )  
 $E = 2 * 4 = 8$
3. Subtract the expected weight form the length (64 - 8 = 58)
4. The order = round (58 / 20) nucleotides in the path = 3<sup>rd</sup> vertex.

Finally, based on previous example we can contribute that our proposed DNA algorithm is visible to solve shortest Hamiltonian path problem.

VII. DISCUSSION AND CONCLUSION

DNA computing is a promising method for unconventional computation, owing to its merits of massive parallelism and efficiency in NP problem solving. One of the most challenging topics in the field of molecular or DNA computing is how to obtain an efficient degree of spatial complexity in “manufacturing” the molecules. Here, the word “manufacturing” refers to the tasks for preparing or producing the materials by certain technical methods that will be used to build a molecular computer. Therefore, the major problem for Adleman’s and Lipton’s in DNA computing experiments, is the time involved in extracting and recombining DNA. While DNA processes within the test-tube can take place millions of times per second, extraction processes, whereby individual strands of DNA are manually isolated and spliced, can take several hours and even days, just for the simplest problems. Thus, if we are to apply molecular computing algorithms to the processes of NP- complete problem solving, we really need to obtain a linear order in the space of controlling (i.e., the number of molecules to be controlled) under the condition of linear time complexity. This has led several researchers to conclude that the complexity aspects of DNA algorithms will limit their applicability. However, the research direction in DNA computing field ignores some fundamental biological and computational issues. Such as, the research paper in [3], try to solve this problem by proposed signaling pathways in cells, which is aimed at cutting the cost of building a molecular computer.

Moreover, digital computer provides a way to interact with its processor and memory in such a way that modern programmer simply writes lines of code in some high level language organizing loops, control flow statements and declaration of variables, while silicon-based computers take programmer away from basic operations, DNA computer does not have this ability, to solve a particular problem on DNA molecules one should perform its simplest operations himself spending time in the laboratory.

In this paper, we proposed a new DNA algorithm that solve shortest Hamiltonian path problem. Notably, in literature, there is no research paper proposed to solve this problem yet. For this reasons our proposed solution is original. In other hand, one of the biggest challenges facing the field of DNA computing is that no efficient implementation has been produced for testing, verification, and general experimentation. While Adleman's initial experiment was performed in a lab, many of the subsequent algorithms in DNA computing have never been implemented or tested. For this reason, in future research, we need real experimental DNA algorithm to give measurable and meaningful result.

## REFERENCES

- [1] Adleman, L., "Molecular Computation of Solutions to Combinatorial Problems." *Science*. 266: 1021-1024 (Nov. 11, 1994).
- [2] Donald B., "Molecular Computing", NSF, 1995.
- [3] Jian Q., and Katsunori S., "Signaling-pathway-based molecular computing for efficient 3-SAT problem solving", *Information Sciences*, Vol.161 2004, pp.121-137.
- [4] Suruchi S., Dhiraj B., Yamuna K., "Transforming bases to bytes: Molecular computing with DNA", *Current trend in science*, 2009.
- [5] Max H., and Russell J., "Biomolecular Computing and Programming", *IEEE Transaction on Evolutionary Computation*, Vol. 3, No. 3, 1999.
- [6] Lipton R., "DNA solution of Hard Combinatorial Problems." *Science*, Vol. 268, 1995, pp.542-545.
- [7] Ouyang Q., Kaplan P. D., Liu S., "DNA Solution of the Maximal Clique Problem", *Science*. Vol. 278, 1997, pp.446-449.
- [8] Liu Q., "DNA computing on surfaces", *Nature*, Vol.403, 2000, pp.175-179.
- [9] Adleman L., "Computing with DNA", *Scientific American*, 1998.
- [10] Martyn A., "DNA Computing" Invited article for the *Encyclopedia of Complexity and System Science*, Springer, 2008.
- [11] Arita M., Akio N., Hagiya M., "Improving sequence design for DNA computing", *Japan society for the Promotion of Science*, 2000.
- [12] Arita M., Akio S., Hagiya M., "A Heuristic Approach for Hamilton Path Problem with Molecules", *Japan society for the Promotion of Science*, 1997.
- [13] Shalini R. Vijay S., Naveen H., "Biological computer Model to Solve NP Compleat Problem", *International Journal of Information Technology and Knowledge Management*, Vol 4, No. 1, 2011, pp. 191-194.
- [14] Zuwairie I., Yusei T., Osamu O., "Direct-Proportional Length-Based DNA Computing for Shortest Path Problem", *International Journal of Computer Science & Applications*, Vol. I, No. 1, 2004, pp. 46 - 60.
- [15] Zhenye W., Zhang Q., Dang Y., "The Improvement of DNA Algorithm to the Directed Shortest Hamilton Path Problem", *IEEE Transaction on Evolutionary Computation*, 2009.
- [16] Qing W., Zhi L., Xiu P., Qi S., Hong Z., "DNA algorithm based on incomplete molecule commixed encoding for the shortest path problem", *The 1st International Conference on Information Science and Engineering*, 2009.
- [17] Ajit N., and Spiridon Z., "DNA algorithms for computing shortest paths" *University of Exeter press*, 1998.
- [18] Lovgren S., "Computer Made from DNA and Enzymes". *National Geographic*. Retrieved 2009.
- [19] Yaakov B., Binyamin G., Uri B., Rivka A., Ehud S., "An autonomous molecular computer for logical control of gene expression". *Nature journal*, Vol.429, 2004, pp.423-429.
- [20] Shudong W., Wenbin L., Jin X., "A DNA computing model minimal covering by plasmid", *Journal of Huazhong University of Science and Technology*, Vol. 32, 2004, pp.59-61.
- [21] Roweis S, Winfree E, Burgoyne R, Chelyapov NV, Goodman MF, Rothmund PW, Adleman LM, "A sticker-based model for DNA computation", Vol.5, 1998, pp.615-29.