HAMILTONIAN PATH PROBLEM SOLUTION USING DNA COMPUTING

Anna Sergeenko
Saint Petersburg State University
Faculty of
Mathematics and Mechanics
Russia
anna.sergeenko98@gmail.com

Maria Yakunina
Peter the Great St. Petersburg
Polytechnic University
Sechenov Institute of Evolutionary
Physiology and Biochemistry
Russian Academy of Sciences
Russia
yakuninam@gmail.com

Oleg Granichin
Saint Petersburg State University
Faculty of
Mathematics and Mechanics
Russia
o.granichin@spbu.ru

Article history:
Received 05.11.2019, Accepted 20.06.2020

Abstract

In this article we study DNA computing, a method which is based on working with DNA molecules in a laboratory. That approach is implemented in solving one of the most popular combinatorial problem — the Hamiltonian path problem. Related to recent improvements in the biophysics methods, which are needed for DNA computing, we propose to change some steps in the classical algorithm to increase accuracy of this method. The branch-and-bound method, the most popular method which is realized on a computer, is also shown in this paper to compare its performance with the time consumption of DNA computing. The results of that comparison prove that it becomes inefficient to use the branch-and-bound method from the counted number of vertices because of its exponentially growing complexity, while DNA computing works parallel and has linearly growing time consumption.

Key words

Evolutionary computation, Optimization, DNA, Hamiltonian path

1 Introduction

Nowadays new physical and theoretical principles of creating a distinctively new ultra-fast hybrid computer are becoming more and more interesting [Granichin et al., 2015]. That interest is caused by the fact that the existing computers are not able to solve quickly and effectively a number of complex multidimensional problems associated with rapidly changing dynamic processes. There are mainly two barriers [Martyn, 2005] to improve a traditional way of developing computers: the machine architecture and the nature of substrates. On the one hand, miniaturization lead to the increase of the processors speed and the size of he memory. On the other hand, there is a limit in miniaturization. For example, the uncertainty principle makes impossible to measure the state of the atom-size microschemes without changing it because of the quantum interference effect.

There are many problems, which traditional computers nowadays cannot solve if the amount of data is big enough. One of it is the Hamiltonian path problem, a classic optimization problem. It belongs to the class of NP-complete [Garey and Johnson, 1979] and transcomputational problems, so the time required to solve it with the number of vertices over 66 exceeds the lifetime of the universe [Klir, 1991]. At this moment there are several algorithms of solving this problem, such as the brute force method, the branch-and-bound method, the ant colony optimization, the genetic algorithm and others, but most of them have exponentially growing labor intensity, which makes impossible to use all this methods for finding a Hamiltonian path in big graphs.

In 1994 [Adleman, 1994] Adleman proposed one more way of solving Hamiltonian path problem, which is based on DNA computing. It is supposed to be faster than any other algorithms starting with concrete number of vertices. Moreover, its time consumption does not grow exponentially with the number of vertices. The main advantage of DNA computing is its parallelism, which means that all paths are created at the same moment. On the one hand, the original algorithm proposed in 1994 has already gone outdated because its realization meant long and painstaking work in a laboratory. On the other hand, considering the development of the biophysics methods [Sergeenko et al., 2020], DNA computing is now relevant and in demand again. For example, nowadays there are several works related to the DNA-based neural networks [Cherry and Qian, 2018; Qian et al., 2011].

The research of new ways of solving the Hamiltonian path problem and a comparison of their time consumption is both of fundamental interest, since it can provide us with novel principles of building a new computer [Garg et al., 2018; Eshra et al., 2019], and practical ones on their basis: the tasks of building optimal motion patterns, recognizing trajectories, images can be solved using the Hamiltonian path.

The paper is organized as follows: the Hamiltonian path problem is briefly described in Section 2. The branch-and-bound method is shown in Section 3. The DNA computing is illustrated in Section 4. Finally, the comparison of both methods mentioned above are shown in Section 5.

2 Problem Formulation

The description of the Hamiltonian Path problem for a directed graph is the following: given a directed graph $G = (V, E)$ with $|V| = n$ nodes and a start vertex and a stop vertex, the problem asks to compute at least one path, beginning with the first vertex ending with the last, containing all vertices exactly once.

The two graphs that we analyzed in DNA computing are shown in Fig. 1, 2. These graphs consist only of 6 and 12 vertices correspondingly, but it will be shown later that it is enough to demonstrate the algorithm and to prove that the time consumption of DNA computing grows linearly with very low slope angle.

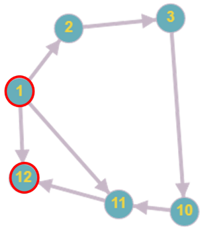

Figure 1. The first network topology.

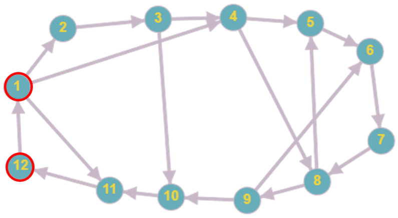

Figure 2. The second network topology.

3 Branch and Bound Method

The branch-and-bound method is supposed to be the most popular method of solving the Hamiltonian path problem. The stages of this algorithm for graph of n vertices are listed below:

1. Assignment the order 1 to the first vertex (according to the task);
2. Checking wherever all vertices are in the path, if not:
   (a) choosing the vertex i from 0 to n;
   (b) finding the nonzero value in the intersection of line i with column j in the incidence matrix (if all values in are zero, take one step back and repeat with excluding that j);
   (c) checking if the vertex j is new to the path;
   (d) assigning the order i to the vertex j;
   (e) back to step 2;
3. Finding if Hamiltonian path exists or not.

If there are a lot of pathways, it does not take long for a computer to evaluate a Hamiltonian path. However, if we take into consideration a situation, when the incidence matrix is sparse, the complexity of branch-and-bound method grows very fast with the number of vertices, because to find at least one Hamiltonian path we have to repeat the steps $2.a - 2.c$ many times. To prove that, the time consumption of the branch-and-bound method was measured for different number of vertices, generating every time a new incidence matrix with fixed sparsity parameter from 0.1 to 0.9 with the step 0.1 (if a random value is larger than sparsity parameter, than the value is 1, otherwise — 0). It occurred that the time consumption of the branch-and-bound method grows exponentially with the number of vertices if only:

$$0.8 \leqslant p \leqslant 0.9,$$

where p is a sparsity parameter of the incidence matrix. The results are shown in Fig. 3

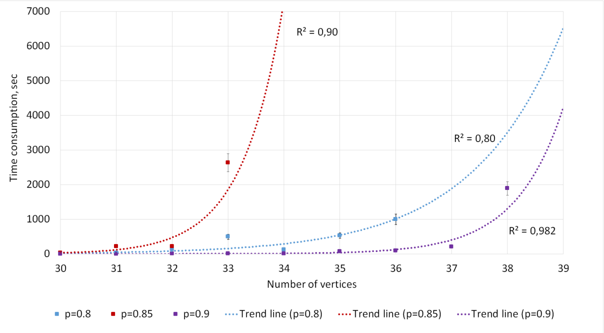

Figure 3. The time consumption of the branch-and-bound method. The dotted lines are the trend lines.

4 DNA Computing

To understand the DNA computing it is essential to know the structure of a DNA molecule. Its monomer is a nucleotide (Fig. 4), which contains a phosphate group, a sugar group and a nitrogenous base. The nucleotides are joined to one another in a chain by covalent bonds. The four types of nitrogenous bases are adenine (A), thymine (T), guanine (G) and cytosine (C). There are many DNA sequences due to many variants of nitrogenous bases order. What is more important for DNA computing, the nitrogenous bases of the two separate strands are bound together according to base pairing rule called *complementarity* with hydrogen bonds to make double-stranded DNA (Fig. 5). The complementarity says that A is always opposite to T and G is always opposite to C (Table 1). That rule can be justified by chemical structure of each nitrogen base [Watson and Crick, 1953].

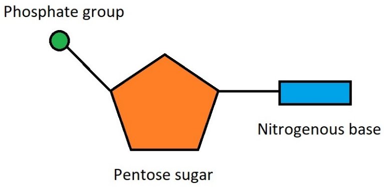

Figure 4. The structure of a nucleotide.

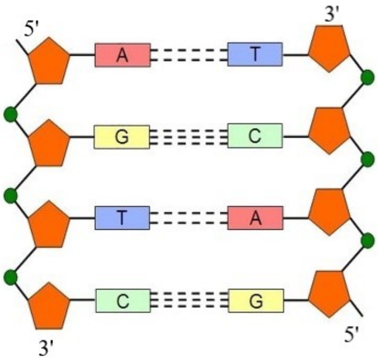

Figure 5. The DNA structure.

Table 1. The complementarity rule.

A	T	G	C
T	A	C	G

All DNA sequences have the direction. The ends of a single stranded DNA are called $5'$ and $3'$ ([Martyn, 2005]). In a nucleic acid double helix, the direction of the nucleotides in one strand is opposite to their direction in the other strand.

Different laboratory methods allow us to work with DNA molecules. For example, we are able to attach to them proteins, to split double stranded DNA on two separate strands (by heating) or to do reverse action (with the help of the special enzyme *ligase*). The steps and the results of DNA computing are shown below. All processes except for the first stage were made in a biophysics laboratory.

4.1 Association of Vertices in Graph with DNA

Each vertex in the graph was associated with a random 20-mer sequence of DNA. Afterwards, we associated the available pathways with DNA sequence in a such way: we took the second complement half of the starting vertex and the first complement half of the finishing vertex. For example, if the first vertex has the DNA code

$$5' - TCAGTACCAG \quad TACAGTCACA - 3'$$
$$(3' - AGTCATGGTC \quad ATGTCAGTGT - 5'),$$

where A — adenin, T — timin, G — guanin, C — citosin, the sixth vertex has the DNA code

$$5' - TAGGTATGCT \quad CAGATAAAGG - 3'$$
$$(3' - ATCCATACGA \quad GTCTATTTCC - 5'),$$

and there is a path from the first vertex to the sixth, then that path is coded by

$$3' - ATGTCAGTGT \quad ATCCATACGA - 5'.$$

To implement that stage it is important to take care of the direction of the lines. We repeated the same procedure with every available pathway.

4.2 Ligation

After associating every vertex and every pathway with DNA, we synthesized all these molecules in a laboratory to be able to work with them. Then we mixed the created molecules in a single ligation reaction. In this stage due to complementarity and the action of the ligase all available pathways were created in one probe at the same moment. It is important to mention, that all paths, both short and long, Hamiltonian and non-Hamiltonian, were made parallel. The example of one short molecule is shown in Fig. 6.

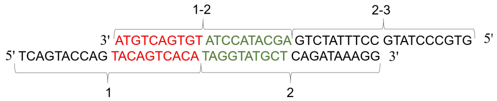
1-2
2-3
3' ATGTCAGTGT ATCCATACGA GTCTATTTCC GTATCCCGTG 5'
5' TCAGTACCAG TACAGTCACA TAGGTATGCT CAGATAAAGG 3'
1
2

Figure 6. The example of one molecule after ligation.

4.3 Amplification

Then we put an amplification reaction, so that all molecules encoding the first and the last vertices according to the task were copied. That was made by polymerase chain reaction (Fig. 7), a method which is widely used in molecular biology to amplify DNA sequences. It is implemented in three stages:

1. splitting double stranded DNA on two single stranded DNA;
2. annealing of small special fragments called *primers* to the single stranded DNA from the previous step;
3. annealing of nucleotides to the single stranded DNA with primers according to the complementary rule.

These stages are repeated for several times ($\sim$ 30). The switching of the stages is done by changing the temperature of the mixture.

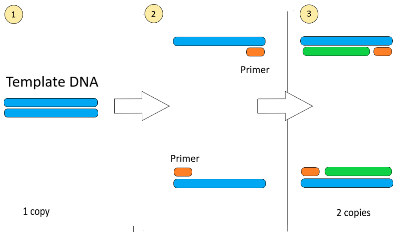

Figure 7. Polymerase chain reaction.

To do PCR we had to make a mixture od DNA molecules, primers, nucleotides and polymerase, so after that we set ethanol precipitation to purify DNA.

4.4 Gel-electrophoresis

Later we had to select molecules, which encode the right number of vertices. That stage was done by setting the gel-electrophoresis. It is a biophysics method, that is made to sort molecules according to their length. DNA molecules are negatively charged, so if they are put in gel with pores while the electric field is turned on, molecules start to move. The shorter the DNA, the faster it gets to anode. There is always a DNA sequence with known length (a marker), so in order to get the length of a molecule you need to compare it with a marker. The picture which we got after electrophoresis is shown in Fig. 8. It is well seen that there are a lot of molecules with different lengths. After making that picture we carved the gel fragments which contained 120 and 240 nucleotides (6 and 12 vertices in the graphs, each has 20 nucleotides length) and again did the gel-electrophoresis to see what was carved. As it is shown in Fig. 9, for the graph with 6 vertices only the right fragment was left (120 nucleotides), while the next line for the graph with 12 vertices apart from 240 nucleotide fragments contained also other DNA molecules (220 nucleotides). However, during next steps these odd fragments should have been disappeared.

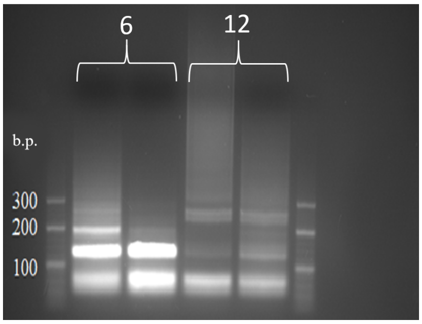

Figure 8. The electrophoresis before carving. The first line and the sixth lines are the markers. The lines two and three are the result for the graph with 6 vertices, four and five — 12 vertices.

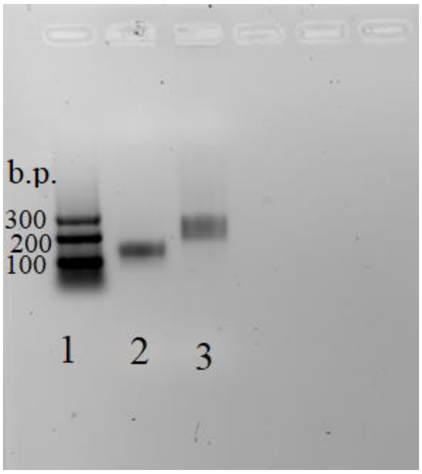

Figure 9. The electrophoresis after carving. The first line is the marker. The lines two is the result for the graph with 6 vertices, three — 12 vertices.

To implement the next stage we had to heat the carved fragments of gel and extract DNA molecules. That was made with a special kit.

4.5 Biotin-Streptavidin System

We had to check wherever all vertices were included in the molecules, because otherwise it is not a Hamiltonian path. That was made by using biotin-streptavidin magnetic beads system (Fig. 10). Firstly, we generated single-stranded DNA from the double-stranded DNA product (by heating on 80°). Secondly, we incubated the single-stranded DNA with the complement sequence of the first vertex conjugated to magnetic beads (room temperature). Only those single-stranded DNA molecules that contained the sequence of the first vertex annealed to the bound and after were retained (80°). This process was repeated with all other complement vertices.

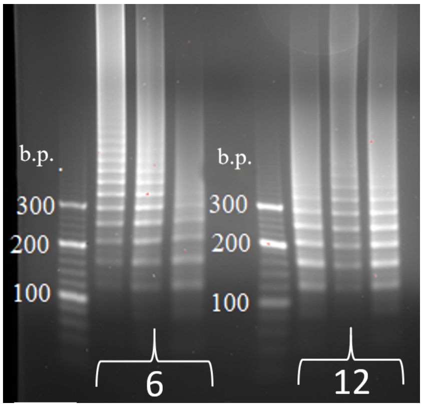

Figure 12. The results of the gel-electrophoresis.

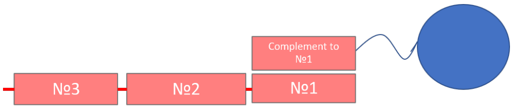

Figure 10. Magnetic system.

4.6 Denaturing Gel-Electrophoresis

To check if any molecules left in the probe we did denaturing gel-electrophoresis, because after purification with biotin-streptavidin system only single-sanded DNA could have been left. Denaturing gel-electrophoresis is a specific type of gel-electrophoresis which is run under conditions that disrupt the natural structure of the DNA, causing it to unfold into a linear chain. To do that we included urea in the gel. The results of that process are shown in Fig. 11.

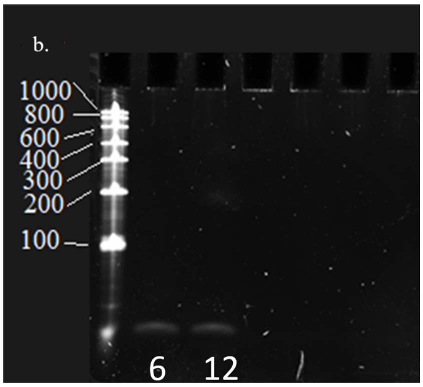

Figure 11. The results of the denaturing gel-electrophoresis.

After implementing the previous stage and according to the topology of the graphs, which were analyzed using DNA computing, we expected to get for both graphs the answer that the Hamiltonian path exists. However, there were no Hamiltonian Path molecules left in the both probes. That might have appended because nucleotides with biotin, which are well seen in Fig. 11 retained from streptavidin beads because of the heating of the probes. In [Holmberg et al., 2005] it is said that the retaining temperature of the biotin-streptavidin is less that 80°, which means that after each step of biotin-streptavidin purification those free biotin nucleotides interfered the conjugation of single nucleotides with biotin-streptavidin system. However, Fig. 12 shows the results of doing PCR again to increase the concentration of DNA and visualizing it on gel-electrophoresis. If there were no Hamiltonian molecules in the probe, then we would not have gotten long molecules in the PCR-product. That means that the Hamiltonian path exists, but the concentration of such molecules is very low. In the future works we are planning to change the stage called biotin-streptavidin system on sequencing to improve its accuracy and to enable reading the path.

5 The Comparison of Two Methods

The Hamiltonian path problem was solved using two methods. It was shown in Section 3 that the time consumption of branch-and-bound method grows exponentially if the sparsity parameter of the incidence matrix lies between 0.8 and 0.9.

The average time required to solve the Hamiltonian path problem by using the branch and bound method is shown in Table 2. The time required to solve the Hamiltonian path problem is shown in Table 3.

There is only one stage (biotin-streptavidin system) in DNA computing, which time consumption depends on the number of vertices, so the complexity of DNA computing is not fixed but is linear. The comparison of two methods is illustrated in Fig. 13.

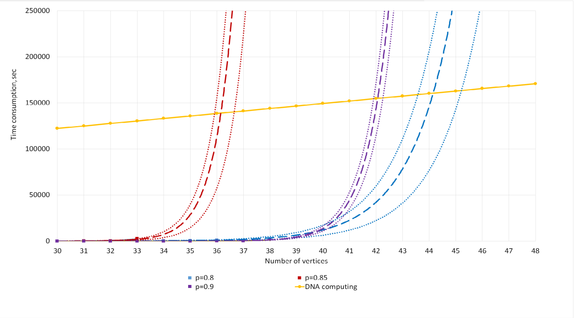

Figure 13. The comparison of the two methods. The dotted lines are the trend lines with the approximation error (the significance level is 0.05).

For such sparsity parameters as 0.85, 0.9, 0.8 and if number of vertices in a graph exceeds 37, 43, 45 correspondingly, the DNA computing works faster that the branch-and-bound method. However, if the sparsity parameter is not between 0.8 and 0.9, then the DNA computing will be implemented much longer compared to

Table 2. Time for solving the Hamiltonian path problem (branch-and-bound method, p=0.9)

Number of vertices	34	35	36	37	38
Time, seconds	12.16	63.93	89.89	201.05	1887.56

Table 3. Time for solving the Hamiltonian path problem (DNA computing)

Step	1	2	3	4	5	6	7	8	Sum
Time, minutes	240	69	60	60	15	50n+5	30	210	689+50n

any computer method. It is important to mention that on a practical basis there is no simple way of measuring the sparsity parameter of the incidence matrix. Moreover, using DNA computing enables us to solve the Hamiltonian path problem with larger number of vertices than the branch-and-bound method because of its linearly growing time consumption. Despite the fact that the time needed for DNA computing grows linearly with the number of vertices, the volume of oligonucleotides tends to grow exponentially. However, as it was discussed before, DNA computing is suitable for finding solutions in sparse graphs, so till certain number of nodes the amount of material does not pose a problem.

6 Conclusion

In this paper we present the algorithm of solving Hamiltonian path problem using DNA computing. Also we compare this method with the branch-and-bound method and show why DNA computing is more efficient if the number of vertices in graph exceeds 43. We explained only basic principles of computing using DNA molecules and provide information how this method can be modified to improve its accuracy. DNA computing can be also used to solve other NP-complete problems, such as, for example, the travelling salesman problem. We strongly believe that in the future it is going to be possible that a programmer by clicking a button activates the processes in a tube, such as preliminary compiling (associating vertices with DNA), the calculation process (that is the stage called a new evolutionary computation, because the nature works) and the results output (sequencing).

Acknowledgements

This work was supported by Russian Foundation for Basic Research (project 20-01-00619).

References

Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. *Science, New Series*, **266** (5187), pp. 1021–1024.

Cherry, K. and Qian, L. (2018). Scaling up molecular pattern recognition with dna-based winner-take-all neural networks. *Nature*, **559** (7714), pp. 370–376.

Eshra, A., Shah, S., Song, T., and Reif, J. (2019). Renewable dna hairpin-based logic circuits. *IEEE Transactions on Nanotechnology*, **XX** (X).

Garey, M. and Johnson, D. (1979). *Computers and intractability: A guide to NPCompleteness*. Freeman and Co.

Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., and Reif, J. (2018). Renewable time-responsive dna circuits. *Small*, **14** (33).

Granichin, O., Volkovich, V., and Toledano-Kitai, D. (2015). *Randomized Algorithms In Automatic Control And Data Mining*, vol. 67 of *Intelligent Systems Reference Library*. Springer.

Holmberg, A., Blomstergren, A., Nord, O., Lukacs, M., Lundeberg, J., and Uhlén, M. (2005). The biotin-streptavidin interaction can be reversibly broken using water at elevated temperatures. *Electrophoresis*, **26**, pp. 501–510.

Klir, G. (1991). *Facets of systems science*. Springer.

Martyn, A. (2005). *Theoretical and Experimental DNA Computation*. Springer.

Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with dna strand displacement cascades. *Nature*, **475** (7536), pp. 368–372.

Sergeenko, A., Granichin, O., and Yakunina, M. (2020). Hamiltonian path problem: The performance comparison deoxyribonucleic acid computing and the branch-and-bound method. *Journal of Physics: Conference Series*, **1536**.

Watson, J. and Crick, F. (1953). A structure for deoxyribose nucleic acid. *Nature*, **171** (346), pp. 737—-738.