Hamiltonian path problem: the performance comparison
deoxyribonucleic acid computing and the branch-and-bound method

A.N. Sergeenko¹, O.N. Granichin², and M.V. Yakunina³

¹ Student, Saint Petersburg State University, Institute of Problems in Mechanical Engineering, Russian
Academy of Sciences (RAS), Saint Petersburg, Russia
² Doctor of Physics and Mathematics, Professor, Saint Petersburg State University, Institute of Problems in Mechanical Engineering, Russian Academy of Sciences (RAS), Russia
³ Candidate of Biological Sciences, Peter the Great St. Petersburg Polytechnic University, Sechenov institute of evolutionary physiology and biochemistry, RAS, St. Petersburg; Russia

E-mail: anna.sergeenko98@gmail.com

Abstract. In this article different approaches to one of the most popular combinatorial problem — the Hamiltonian path problem — are illustrated and compared between each other. It is shown that it becomes inefficient to use branch-and-bound method, the most popular method which is realized on a computer, from the counted number of vertices because of its exponentially growing complexity, one more algorithm which is based on working with deoxyribonucleic acid (DNA) molecules in a laboratory is analysed. That method works parallel and has linearly growing time consumption. Due to the improvements in the biophysics methods, which are needed for DNA computing, that algorithm became much faster than it was several years ago and it is now possible to add some new stages in DNA computing, which are shown in this paper.

Introduction
Nowadays new theoretical principles of creating a distinctively new ultra-fast hybrid computer are becoming more and more interesting [6]. That interest is caused by the fact that the existing computers are not able to quickly and effectively solve a number of complex multidimensional problems associated with rapidly changing dynamic processes. One of this problems is the Hamiltonian path problem, a classic optimization problem. It belongs to the class of NP-complete [4] and transcomputational problems, so the time required to solve it with the number of vertices over 66 exceeds the lifetime of the universe [7]. At this moment there are several algorithms of solving this problem, such as the brute force method, the branch-and-bound method, ant colony optimization, the genetic algorithm and others, but most of them have exponentially growing time consumption, which makes the use of all this methods for finding a Hamiltonian path in big graphs almost impossible.

In 1994 [1] Adleman proposed one more way of solving Hamiltonian path problem, which is based on DNA computing and means working with DNA molecules. It is supposed to be faster than any other algorithms starting with concrete number of vertices. Moreover, its performance 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. That is the main reason why DNA computing is very popular now. For example, nowadays there are several works related to the DNA based neural networks [2].
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 [3, 5, 8], and practical ones on their basis: the tasks of building optimal motion patterns, recognizing trajectories, images are solved using the Hamiltonian path.

1. 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 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 [3, 5, 8], and practical ones on their basis: the tasks of building optimal motion patterns, recognizing trajectories, images are solved using the Hamiltonian path.

1. 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 graph that we analyzed in DNA computing is shown in Fig. 1. The incidence matrix of that graph is shown in Fig. 2. That graph consists only of 12 vertices, but it will be shown later that it is enough to prove that the time consumption of DNA computing grows linearly with very low slope angle so the time required is almost constant.

2. Methods
The branch-and-bound method is the most popular computer method to solve the Hamiltonian path problem. It analyzes the incidence matrix of the path and gives the answer wherever the Hamiltonian path exists or not. The stages of this algorithm for graph of \( n \) vertices are:
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) repeat the 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 1 Hamiltonian path we have to repeat the steps 2.a-2.c many times.

To understand the DNA computing it is essential to know the structure of a DNA molecule. It consists of nucleotides, each containing a phosphate group, a sugar group and a nitrogen base (Fig.3). The four types of nitrogen bases are adenine (A), thymine (T), guanine (G) and cytosine (C). There are many DNA sequences due to many variants of order of these bases. What is more important, DNA molecules usually form double helix: one strand stands opposite to another holding by hydrogen bonds, and there is a very important rule called complementarity: A is always opposite to T, G is always opposite to C, because in this case a DNA molecule is in the state with the smallest free energy. The complementarity rule can be justified by chemical structure of each nitrogen [9].
Firstly, each vertex in the graph needs to be associated with a random 20-mer sequence of DNA. Afterwards, we need to write down and associate the available pathways in a such way: we take the second complement half of the starting vertex and the first complement half of the finishing vertex. It is important to take care of the direction of the lines. We repeat the same procedure with every available pathway. For example, if the first vertex has the DNA code AAAAAAAAAA TATATATATA; where A — adenin, T— timin, the sixth vertex has GCGCGCGCGC CCCCCCCCCC; where G—guanin, C—citosin and there is a line from the first vertex to the sixth, then the pathway is coded by ATATATATAT CGCGCGCGCG.

After associating every vertex and every pathway with DNA, we synthesize all these molecules in a laboratory to be able to work with them. Then we mix the created molecules in a single ligation reaction. Due to complementarity and the action of ligases all available pathways are being created in one probe at the same moment, only if the concentration of DNA molecules is enough. What is important, all paths, both short and long, Hamiltonian and non-Hamiltonian, are created parallel. Then we put an amplification reaction, so that all molecules encoding the first and the last vertices according to the task are copied. That can be made by polymerase chain reaction (Fig. 4), a method which is widely used in molecular biology to amplify DNA sequences. Afterwards, all that is left is to take out the DNA molecules, which encode Hamiltonian path.

We have to find molecules, which encode the right number of vertices. That stage can be made by setting the electrophoresis. It is a biophysics method, that is made to sort molecules according to their length. DNA molecules are negatively charged, so if you put them in a gel with pores and turn on the electric field, they start to move. The shorter the molecule, the closer it gets to anode. Usually there is a DNA sequence with known length (a marker), so in order to get how long is your molecule you need to compare them. The picture which we got after electrophoresis is shown in Fig. 5. It is well seen that there are a lot of molecules with different lengths. After making that picture we carve the gel fragments which contained 240 nucleotides (12 vertices, each has 20 nucleotides length) and again did the electrophoresis to see what was carved. As it is shown in Fig. 6, the lines 4 and 5 contain the molecules with the right length (240), while the lines 2, 3 have another fragments, which were carved accidentally (120).

Afterwards, we have to check, wherever all vertices are included in the molecules, otherwise it is not a Hamiltonian path. That can be made by using biotin-avidin magnetic beads system. Firstly, we generate single-stranded DNA from the double-stranded DNA product. Secondly, we incubate the single-stranded DNA with complement sequence of the first vertex conjugated to magnetic beads. Only those single-stranded DNA molecules that contained the sequence of the first vertex anneal to the bound and are retained. This process is repeated with all other complement vertices. In the end we get molecules, which encode a path starting and finishing in the fixed vertices, have the right length and contain all vertices. If any DNA molecules are left in a probe, then that means that the Hamiltonian path exists. That can be checked with use of electrophoresis, the method which was discussed before.
Fig. 5. The electrophoresis before carving. The first line is the marker (the brightest straps are 100, 200, 300 nucleotides from the bottom to the top) and next 3 are the PCR products.

Fig. 6. The electrophoresis after carving. The first line is the marker (the brightest straps are 100, 200, 300 nucleotides from the bottom to the top), the next 4 lines are the DNA molecules, which were carved from the gel.

3. The comparison of two methods

The Hamiltonian path problem was solved using two methods. 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 happened so that the time consumption of the branch-and-bound method grows exponentially with the number of vertices if only: $0.8 \leq p \leq 0.9$, where $p$ is a sparsity parameter of the incidence matrix. There is only one stage (affinity purification) 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 are shown in Fig. 7.

Fig. 7. The time consumption of DNA computing and the branch-and-bound method (the dotted lines are the approximation error with the significance level 0.05).
Conclusion
In this paper we presented the algorithm of solving Hamiltonian path problem using DNA computing. Also we compared this method with the branch-and-bound method and showed 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. 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).

This work was supported by Russian Science Foundation (project 16-19-00057)

REFERENCES
[1] Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. Science, New Series, 266(5187):1021–1024.
[2] Cherry, K. and Qian, L. (2018). Scaling up molecular pattern recognition with dna-based winner-take-all neural networks. Nature, 559(7714):370–376.
[3] Eshra, A., Shah, S., Song, T., and Reif, J. (2019). Renewable dna hairpin-based logic circuits. IEEE Transactions on Nanotechnology, XX(X).
[4] Garey, M. and Johnson, D. (1979). Computers and intractability: A guide to NPCompleteness. Freeman and Co.
[5] Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., and Reif, J. (2018). Renewable time-responsive dna circuits. Small, 14(33).
[6] Granichin, O., Volkovich, V., and Toledano-Kitai, D. (2015). Randomized Algorithms In Automatic Control And Data Mining, volume 67 of Intelligent Systems Reference Library. Springer.
[7] Klir, G. (1991). Facets of systems science. Springer.
[8] Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with dna strand displacement cascades. Nature, 475(7536):368–372.
[9] Watson, J. and Crick, F. (1953). A structure for deoxyribose nucleic acid. Nature, 171(346):737—-738.