

OVERVIEW OF DNA COMPUTING

Siddhartha Sankar Biswas

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jamia Hamdard
New Delhi, INDIA
Ssbiswas1984@gmail.com

Abstract— This paper tries to explain complex DNA Computing in an technically easy manner. As this field is still emerging, it gives an immense oppurtinity for research. This paper tries to mainly focus on basics of DNA computing in Adleman's Experiments.

Index Terms— DNA, Algorithms, Hamiltonian Path, Graph, Topology.

I. INTRODUCTION

All Since the field's genesis in 1994, the practice of DNA computing has grown significantly in both the number of problems that it might be used to solve and our understanding of its power after not so successful attempt of quantum computers. It is one of alternate methods of computation have been formalized and, to a lesser extent, implemented to solve problems [1,2,3,4]. One recent example of such a method is what is known as DNA computing, which attempts to solve these problems by taking advantage of the storage capacity of our genetic code.

It is the purpose of this paper to describe DNA computing and emerging computers (Future's Computers) from Adleman's initial experiments to the current state of the field, in order to get perspective on it's future viability. Biochemical "nano-computers" already exist in nature; they are manifest in all living things. But they're largely uncontrollable by humans. We cannot, for example, program a tree to calculate the digits of pi.

The idea of using DNA to store and process information took off in 1994 when a California scientist first used DNA in a test tube to solve a simple mathematical problem[5,6,7]. Think of DNA as software, and enzymes as hardware. Put them together in a test tube. The way in which these molecules undergo chemical reactions with each other allows simple operations to be performed as a byproduct of the reactions. The scientists tell the devices what to do by controlling the composition of the DNA software molecules. It's a completely different approach to pushing electrons around a dry circuit in a conventional computer.

To the naked eye, the DNA computer looks like clear water solution in a test tube. There is no mechanical device. A trillion bio-molecular devices could fit into a single drop of water. Instead of showing up on a computer screen, results are analyzed using a technique that allows scientists to see the length of the DNA output molecule. Once the input, software, and hardware molecules are mixed in a solution it operates to completion without intervention.

II. ADLEMAN'S EXPERIMENTS

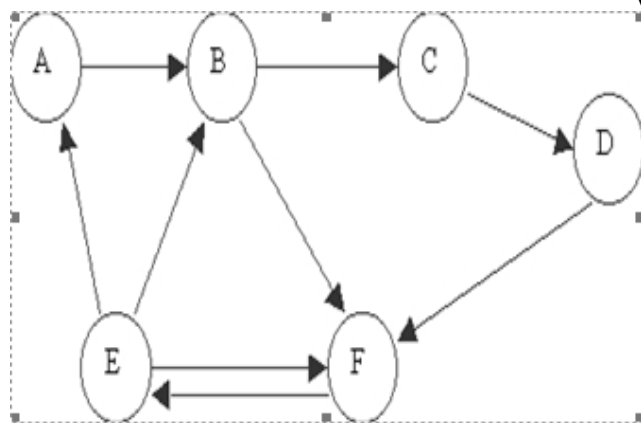
Wherever The field of DNA computing is generally considered to have begun with Leonard Adleman's 1994 experiment involving the Hamiltonian Path (HP) problem. Simply stated, the HP problem is to determine whether or not a graph, with fixed starting and ending vertices, has some sequence of steps by which every vertex in the graph is visited exactly once. Formally, an instance of HP takes the form (V, E, s, d) , where V and E are the sets of vertices and edges that define the graph's topology and s and d are the start and destination vertices.

Adleman found a DNA algorithm by which one could determine, in linear time, whether or not such an instance belonged to HP (membership meaning that there is some path satisfying the criteria above) [8,9,10]. The choice of this problem is rather important in light of the fact that HP is known to be NP-complete, which is to say that any problem whose solution can be verified in polynomial time may be reduced to HP by some polynomial time algorithm. This is a truly remarkable result, as it shows that all problems in NP can be solved, by reduction, in polynomial time (due mostly to the reduction) by a DNA-based computer.

In order to generate DNA sequences that represent all paths through the graph, one must first decide upon a DNA-based representation for the vertices and edges that comprise it. The design of the edges will follow the design of the vertices, so we choose to designate unique DNA strands for each vertex.

The actual length of these strands depends, of course, on the size of the set V , and is chosen such that no two strands will have long common sub-strings. One could represent each of n vertices by a DNA strand of length $\log_4(n)$ (rounded toward infinity). With this representation in mind, we create test tubes that have many multiples of the strands representing each edge and each vertex. Once these tubes are mixed together, legal paths through the graph are generated by the mutual attraction of strings and their complements by hydrogen bonding.

For example, take the graph G , as shown in figure 1 below. Since this example has six vertices and the DNA alphabet only contains the letters A, T, C, and G, it is necessary to have each vertex represented by at least two characters.



ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put applicable sponsor acknowledgments here; DO NOT place them on the first page of your paper or as a footnote.

Figure 1 : Graph G

We expand this to eight characters in order to avoid the problem of long common sub-strings, and denote each vertex by a DNA strand as listed in Table I. The edges, furthermore, are represented as in Table II.

TABLE I. DNS strands of Vertices Notice that the structure of th

Vertex	DNA Strand
A	ATCGAGCT
B	TGGACTAC
C	GAGACCAG
D	CGATGCAT
E	AGCTAGCT
F	GCTAGACT

TABLE II. DNS strands of Edges

Edge	DNA Strand
AB	TCGAACCT
BC	GATGCTCT
CD	GGTCGCTA
DF	CGTACGAT
FE	CTGATCGA
EF	TCGACGAT
EA	TCGATAGC
EB	TCGAACCT
BF	GATGCGAT

e strands representing each edge follows from the strands representing the vertices as described above. Adleman's initial experiment was performed on an instance of HP with seven

vertices and relatively few edges. The algorithm successfully discovered the Hamiltonian path and was considered a success.

In his concluding remarks, Adleman suggested that larger, more useful, instances of this problem could be solved simply by increasing both the length of the strands representing each edge and vertex, as well as increasing the number of copies of each strand used in the experiment.

This initial success created quite a bit of clamor in the computing community. Researchers began defining more and more algorithms to solve hard problems using DNA computing while others began to think about what a true "DNA computer" would look like.

Furthermore, theoreticians began to question the true computing power of this new method, and whether or not it was truly a different paradigm of computation. Steps were also taken to formalize the operations that one could perform on a test tube full of DNA strands and how those steps would relate to traditional notions of Turing computability.

III. Operations on DNA

It may be noted that the types of operations available are a result of the capability of molecular biology rather than the wishes of algorithm designers.

Also note that these algorithms are performed in constant time (ignoring the variability of different technicians, etc) on test tubes which, for the sake of this discussion, may be of arbitrary size. These operations are as follows:

- (i) Merge - This is the simple operation of combining the contents of two test tubes in a third tube.
- (ii) Melt - Melting is the inverse operation of annealing. By heating the contents of a tube, double-stranded DNA sequences are denatured, or separated into its two single-stranded parts.
- (iii) Separation by length - 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.** }
- (iv) Copying/Amplification -Copies are made of DNA strands in a test tube.The strands to be copied must have known sequences at both the beginning and end in order for this operation to be performed.
- (v) Append - This process makes a DNA strand longer by adding a character or strand to the end of each sequence.
- (vi) Detect - It is also possible to analyze a test tube in order to determine whether or not it contains at least one strand of DNA.

IV. SOME IMPORTANT RESULTS IN DNA COMPUTING

- (i) Let us suppose a DES has a 56-bit key, meaning that the initial solution would have at least 256 strands.
- (ii) DNA operations are performed on the test tube that mirror the DES encryption of that plaintext with the key specified by the value of the strand.
- (iii) At the end of the operation, which is also linear in time, the resultant test tube contains all (key, cyphertext) pairs derived from the known plaintext
- [2] A. Condon, R. Corn, and A. Marathe "Combinatorial DNA Word Design". DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Pages75-90 (2000). Available at <http://corninfo.chem.wise.edu/writings/DNAcomputing>.
- [3] L.M.Adleman, "Molecular computation of solutions to combinatorial problems" Sciences,Vol.266, No.5187, 1994, pp. 1021-1024
- [4] L.Kari, "From micro-soft to bio-soft: Computing with DNA", Biocomputing and emergent computation: Proceedings of BCEC97. World Scientific 1997, Skovde, Sweden, 1997, pp. 146 - 164.
- [5] G. Rozenberg and A. Salomaa, "DNA computing: New ideas and paradigms," Lecture Notes in Computer Science (LNCS), Springer-Verlag, Vol. 7, 2006, pp. 188 - 200.
- [6] A.Narayanan and S.Zorbalas, "DNA algorithm for computing shortest paths," In J.R. Koza (ed.), Proceedings of the Genetic Programming, Morgan Kaufman, 1998, pp. 718 - 723.
- [7] Z.Ibrahim, Y. Tsuboi, O.Ono and M.Khalid, "Directproportional length-based DNA computing for shortest path problem," International Journal of Computer Sciences & Applications, Vol.1, No.1, 2004, pp. 46-60.
- [8] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [9] R.J. Lipton, "DNA solution of hard computational problems," Sciences, Vol. 268, No.5210, 1995, pp. 542 - 545.
- [10] M.S. Muhammad, Z. Ibrahim, O.Ono and M.Khalid, "Application of length-based DNA computing for complex scheduling problem," International Journal of Information Technology, Vol.12, No.3, 2006, pp. 100- 110.

V. CONCLUSION

DNA computing research is going so fast that its potential is still emerging. This is an area of research that leaves the science fiction writers struggling to keep up. Despite various issues involved, it is certainly possible that in some instances a DNA-based computing system may prove to be the best solution.

DNA computing will become a replacement for electronic computing in the near future. Given the high cost and required space, it is hard to imagine the use of a DNA-based computer in many of the places where computers exist today, though in future who knows, every house on the planet might be having a DNA computer system installed for multiple purposes.

REFERENCES

- [1] Md. Ezaz Ahmed, Sumit Pandey "DNA Computing and Emerging Computers" National Seminar on Knowledge Based Intelligent Engineering System,ITM Gurgaon.