Emerging Paradigms Of Dna Based Computation

Dr. Rajendra Singh*
Prof. Ajay Malpani**

1. Introduction: DNA computing is a novel technology that seeks to capitalize on the enormous informational capacity of DNA, biological molecules that can store huge amounts of information and are able to perform operations similar to that of a computer, through the deployment of enzymes, biological catalysts that act like software to execute desired operations. The appeal of DNA computing lies in the fact that DNA molecules can store far more information than any existing conventional computer chip. Also, utilizing DNA for complex computation can be much faster than utilizing a conventional computer, for which massive parallelism would require large amounts of hardware, not simply more DNA. Scientists have found the new material they need to build the next generation of microprocessors. Millions of natural supercomputers exist inside living organisms, including your body. DNA (deoxyribonucleic acid) molecules, the material our genes are made of, have the potential to perform calculations many times faster than the world's most powerful human-built computers. DNA might one day be integrated into a computer chip to create a so-called biochip that will push computers even faster. DNA molecules have already been harnessed to perform complex mathematical problems. While still in their infancy, DNA computers will be capable of storing billions of times more data than your personal computer.

The practical possibility of using molecules of DNA as a medium for computation was first demonstrated by Adleman in 1994. In 1994, Leonard Adleman took a giant step towards a different kind of chemical or artificial biochemical computer. He used fragments of DNA to compute the solution to a complex graph theory problem. Adleman's method utilizes sequences of DNA's molecular subunits to represent vertices of a network or `"graph". Thus, combinations of these sequences formed randomly by the massively parallel action of biochemical reactions in test tubes described random paths through the graph. Using the tools of biochemistry, Adleman was able to extract the correct answer to the graph theory problem out of the many random paths represented by the product DNA strands. Adleman's primary intention was to prove the feasibility of bio molecular computation but his work also gave an indication that the emergence of this new computational paradigm could provide an advantage over conventional electronic computing techniques. Specifically, DNA was shown to have massively parallel processing capabilities that might allow a DNA based computer to solve hard computational problems in a reasonable amount of time. However, Adleman's brute force search algorithm is not, and was never meant to be, a practical means of solving such problems; the volume of material required was found to increase exponentially as the complexity of the problem was increased.

The main idea behind DNA computing is to adopt a biological (wet) technique as an efficient computing vehicle, where data are represented using strands of DNA. Even though a DNA reaction is much slower than the cycle time of a silicon-based computer, the inherently parallel processing offered by the DNA process plays an important role.

*Reader, IMS, DAVV, Indore
**Lecturer, SVIM, DAVV, Indore

This massive parallelism of DNA processing is of particular interest in solving NP-complete or NP-hard problems. It is not uncommon to encounter molecular biological experiments which involve $\mathbf{6\times10^{16}}$ /ml of DNA molecules. This means that we can effectively realize 60,000 TeraBytes of memory, assuming that each string of a DNA molecule expresses one character. The total execution speed of a DNA computer can outshine that of a conventional electronic computer, even though the execution time of a single DNA molecule reaction is relatively slow. A DNA computer is thus suited to problems such as the analysis of genome information, and the functional design of molecules (where molecules constitute the input data).

The concepts of utilizing DNA computing in the field of data encryption and DNA authentication methods for thwarting the counterfeiting industry are subjects that have been surfacing in the media of late. Researchers have been looking at alternatives to the traditional microprocessor design. One of the most interesting and emerging technology is DNA computers. The computing power of a teardrop-sized DNA computer, will be more powerful than the world's most powerful supercomputer.

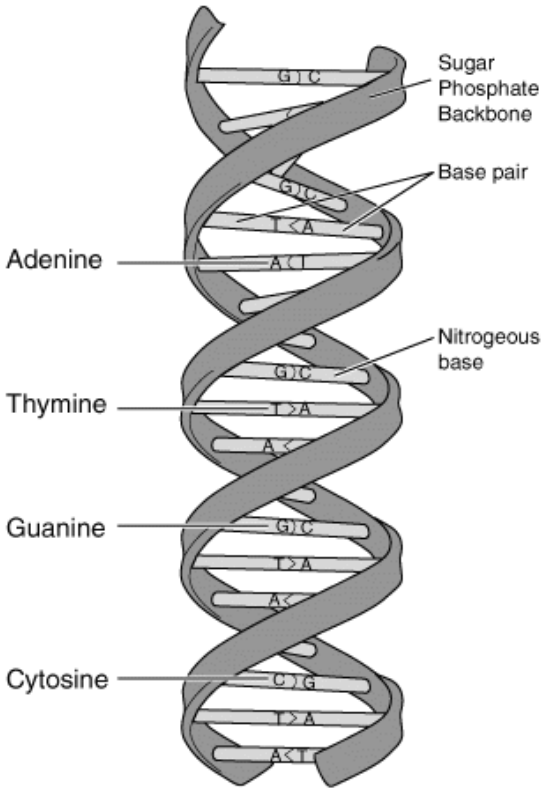
2. The Double Helical Structure of DNA: In 1951, the then 23-year old biologist James Watson traveled from the United States to work with Francis Crick, an English physicist at the University of Cambridge. Crick was already using the process of X-ray crystallography to study the structure of protein molecules. Together, Watson and Crick used X-ray crystallography data, produced by Rosalind Franklin and Maurice Wilkins at King's College in London, to decipher DNA's structure.

1. DNA is made up of subunits which scientists called nucleotides.
2. Each nucleotide is made up of a sugar, a phosphate and a base.
3. There are 4 different bases in a DNA molecule:

   adenine (a purine)
   cytosine (a pyrimidine)
   guanine (a purine)
   thymine (a pyrimidine)

4. The number of purine bases equals the number of pyrimidine bases
5. The number of adenine bases equals the number of thymine bases
6. The number of guanine bases equals the number of cytosine bases
7. The basic structure of the DNA molecule is helical, with the bases being stacked on top of each other

"This (DNA) structure has two helical chains each coiled round the same axis...Both chains follow right handed helices...the two chains run in opposite directions. ..The bases are on the inside of the helix and the phosphates on the outside..."



"The novel feature of the structure is the manner in which the two chains are held together by the purine and pyrimidine bases... The (bases) are joined together in pairs, a single base from one chain being hydrogen-bonded to a single base from the other chain, so that the two lie side by side...One of the pair must be a purine and the other a pyrimidine for bonding to occur. ...Only specific pairs of bases can bond together. These pairs are: adenine (purine) with thymine (pyrimidine), and guanine (purine) with cytosine (pyrimidine)."

"In other words, if an adenine forms one member of a pair, on either chain, then on these assumptions the other member must be thymine; similarly for guanine and cytosine. The sequence of bases on a single chain does not appear to be restricted in any way. However, if only specific pairs of bases can be formed, it follows that if the sequence of bases on one chain is given, then the sequence on the other chain is automatically determined."

Comparison of DNA and Conventional Electronic Computers

DNA-based computers	**Microchip-based computers**
Slow at single operations	Fast at single operations (fast CPUs)
Able to perform billions of ops simultaneously	Can do substantially fewer operations simultaneously
Huge storage capacity	Smaller capacity
Require considerable preparations before	Immediate set up
Chemical deterioration (copy errors)	Vulnerable, easy back up

3. Basic Operations on DNA

Synthesis: Synthesis is a process of designing and restructuring information in DNA sequence form. In DNA computing, designing and synthesizing information in the DNA sequence form is an important process where wrong design might leads to wrong result.

Ligation and hybridization: DNA ligation is a process of joining two single linear DNA fragments together. More specifically, DNA ligation involves creating a phosphodiester bond between 3 -hydroxyl of one nucleotide and the 5 -phosphate of another. Meanwhile hybridization is a process of combining complementary single-stranded nucleic acids into a single molecule. Nucleotides will bind to their complement under normal conditions, so two perfectly complementary strands will bind to each other at the end of the process.

Fig. 1. Droppers for hybridizing

Polymerase chain reaction (PCR): PCR is a process that quickly amplifies the amount of specific molecules of DNA in a given solution using primer extension by a polymerase. DNA polymerases execute several functions including the repair and duplication of DNA. Each cycle of the reaction doubles the quantity of the molecules, giving an exponential growth in the number of operations.

Gel Electrophoresis: Gel electrophoresis is an important technique for sorting DNA strands by their size. Electrophoresis enables charged molecules to move in an electric field, as illustrated in Figure 2. Basically, DNA molecules carry negative charge. Thus, when we place them in an electrical field, they tend to migrate towards a positive pole. Since DNA molecules have the same charge per unit length, they all migrate with the same force in an electrophoresis process. Smaller molecules therefore migrate faster through the gel, and can be sorted according to size (usually agarose gel is used as the medium here). At the end of this process the resultant DNA is photographed, as indicated in Figure 3.

Fig. 2. Electrophoresis

Fig. 3. Camera

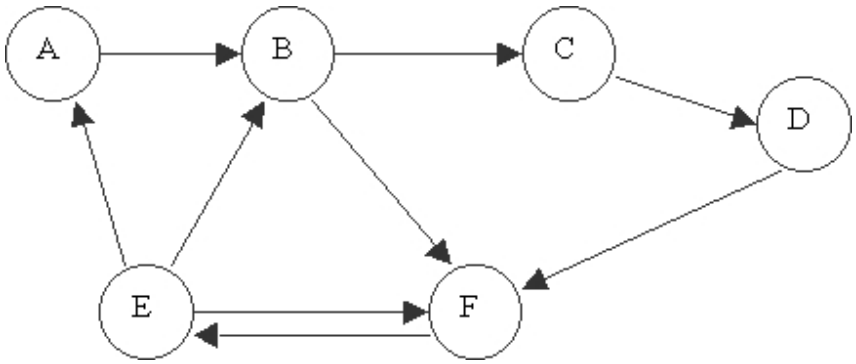
4. Adleman's Experiments: The field of DNA computing is generally considered to have begun with Leonard Adleman's 1994 experiment involving the Hamiltonian Path (HP) problem. Simply stated, the HP problem is to determine whether or not a graph, with fixed starting and ending vertices, has some sequence 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). 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.

Adleman's algorithm for solving HP looks remarkable simple, and can be performed in linear time by the following steps:

1. Generate random paths through the graph
2. Remove those paths that begin with vertices other than *s* and those that end with vertices other than *d*.
3. Remove those paths that visit a number of vertices unequal to the size of the set *V*.
4. Remove those paths that visit some vertex more than once. (Note: this is the step that makes the overall algorithm execute in linear time. This step is a loop, executed once for each vertex, whose body goes about removing paths that contain that vertex more than once.)
5. If there are any paths left, answer "yes," otherwise "no."

We will now discuss in detail how this particular step is performed by a DNA 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. Given some DNA strand, its complement is found by replacing all occurrence of A’s with T’s and vice versa, as well as replacing all instances of C’s with G’s and vice versa. Finally, each edge *a->b* will be represented by a strand (of the same length as the strands representing the vertices) whose first half will be the complement of the second half of the strand representing *a* and whose second half will be the complement of the first half of the strand representing *b*. 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 shown in figure below.



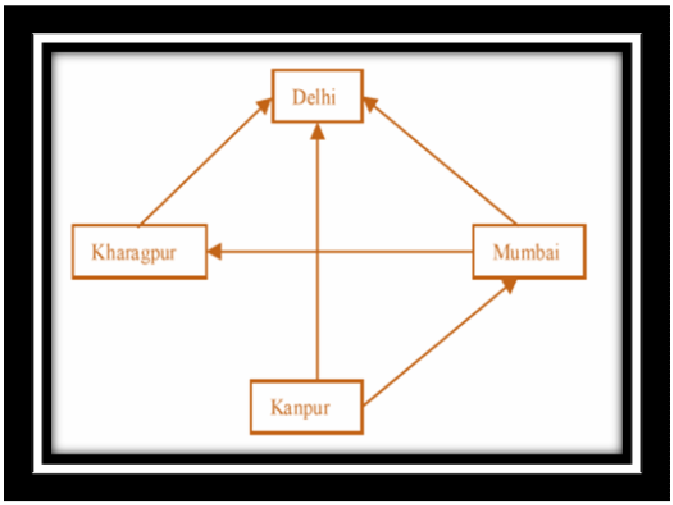
Since this example has six vertices and the DNA alphabet only contains the letters A, T, C, and G, it is necessary to have each vertex represented by at least two characters. We expand this to eight characters in order to avoid the problem of long common sub-strings, and denote each vertex by a DNA strand as listed in table 1. The edges, furthermore, are represented as in table 1. Notice that the structure of the strands representing each edge follows from the strands representing the vertices as described above.

Vertex	DNA Strand	Edge	DNA Strand
A	ATCGAGCT	A -> B	TCGAACCT
B	TGGACTAC	B -> C	GATGCTCT
C	GAGACCAG	C -> D	GGTCGCTA
D	CGATGCAT	D -> F	CGTACGAT
E	AGCTAGCT	F -> E	CTGATCGA
F	GCTAGACT	E -> F	TCGACGAT
		E -> A	TCGATAGC
		E -> B	TCGAACCT
		B -> F	GATGCGAT

Table 1 - Representation of Vertices and Edges from Figure above

We can now see that, given this representation, strands representing vertex *A* in a test tube will tend to pair with strands representing edges of the form *A -> x*. Because of this tendency to pair, one can construct all paths through the graph by simply producing large quantities of each of the strands listed in table 1 and mixing them together. Other environmental conditions must be met so that the pairing process takes place as intended, but those details will be left for later discussions. Once all of the legal paths are constructed, finding a solution (if one exists) to the instance of HP is equivalent to finding a double-stranded DNA sequence that contains exactly one copy of the strands representing each vertex. This search is exactly what is done in steps 2 through 5, and can be performed using methods from molecular biology. Moreover, each of these steps (including the synthesis of the paths) could be performed in linear time and thus the entire algorithm can be executed in linear time.

Example: Consider a map of four cities connected by certain non-stop flights (in the figure). For the example shown here, it is possible to travel directly from Mumbai to Delhi but not vice versa. The goal is to determine whether a path exists that commence at the start city (Kanpur), finish at the end city (Delhi) and pass through each of the remaining cities exactly once.



In DNA computation, Adleman thought of as each city is assigned a DNA sequence (ATCGTCGA for Kanpur) that can be thought of as a first name (ATCG) followed by a last name (TCGA) of the city. DNA flight numbers can then be defined by concatenating the last name of the city of origin with the first name of the destination. In this way we can assign DNA sequences for all cities and flight numbers as follows.

Delhi	TCAGACGA	AGTCTGCT
Kharagpur	CGATCGAT	GCTAGCTA

City name	**DNA Name**	DNA Complement
Kanpur	ATCGTCGA	TAGCAGCT
Mumbai	GTACACTA	CATGTGAT

Flight	DNA Flight Number
Kanpur- Mumbai	TCGAGTAC
Kanpur - Delhi	TCGATCAG
Mumbai - Delhi	ACTATCAG
Kharagpur - Delhi	CGATTCAG
Mumbai – Kharagpur	ACTACGAT
Mumbai - Kanpur	ACTAATCG

We know that each strand of DNA has its Watson-Crick compliment. Thus each city has its complimentary DNA name. Kanpur complimentary name becomes, for instance TAGCAGCT. After working out these encodings, Adleman had the complimentary DNA city names and the DNA flight numbers synthesized. The following algorithm will give the Hamiltonian path for a given graph, if it exists in the graph. Given a directed graph with n vertices, Algorithm:

Step1: Generate a set of random paths through the graph.

Step2: For each path in the set:

a) Check whether that path starts at the start vertex and ends with the end vertex. If not remove the path from the set.

b) Check whether that path passes through exactly n vertices. If not, remove that path from the set.
c) For each vertex, check whether that path passes through that vertex. If not, remove that path from the set.

Step3: If the set is non empty, then report that there is a Hamiltonian path. If the set is empty, report that there is no Hamiltonian path. This is not a perfect algorithm; nevertheless, if the generation of paths is random enough and the resulting set large enough, then there is a high probability that it will give the correct answer. It is this algorithm that Adleman implemented in the first DNA computation.

In the end we can say that,

- Algorithm used by Adleman for the traveling salesman problem was simple. As technology becomes more refined, more efficient algorithms may be discovered.
- DNA computing has a lot of potential Massive parallelism, dense storage
- Can solve NP-complete problems quickly Really, any problem requiring brute-force search of all solutions
- Suited for specific problems
- Getting output takes time

5. Conclusion: DNA computers have tremendous potential to compete with electronic computers, which boasts of superior speeds in computation. A new face in the field of computation is introduced and the possibility of using DNA as a computational tool is high-lightened and described that even a molecular biology laboratory can be made to perform computational operations just like the dry lab or the computer lab, broadening the horizon of computational sciences. Scientists and mathematicians around the world are now looking at the application of DNA computers to a whole range of “intractable” computing problems.
DNA computing can be viewed as ***a manifestation of an emerging new area of science*** made possible by our rapidly developing ability to control the molecular world.

6. References

1. R. P. Feynman, (1961)Miniaturization, New York, Reinhold, pp.282-296, 1961
2. L. M. Adleman, (1994) Molecular computation of solutions to combinatorial problems, Sciences, vol.266, no.5187, pp.1021-1024, 1994.
3. L. Kari,(1997) From micro-soft to bio-soft: Computing with DNA, Biocomputing and Emergent Computation: Proc. of the BCEC97, World Scientific, Skovde, Sweden, pp.146-164, 1997.
4. G. Rozenberg and A. Salomaa,(2006) DNA computing: New ideas and paradigms, Lecture Notes in Computer Science, Springer-Verlag, vol.7, pp.188-200, 2006.
5. A.Narayanan and S.Zorbalas, (1998) DNA algorithm for computing shortest paths, Proc. of the Genetic Programming, Morgan Kaufman, pp.718-723, 1998.
6. Z. Ibrahim, Y. Tsuboi, O. Ono and M. Khalid, (2004) Direct-proportional length-based DNA computing for shortest path problem, International Journal of Computer Sciences & Applications, vol.1, no.1, pp.46-60, 2004.
7. M. Yamamoto, N. Matsuura, T. Shiba, Y. Kawazoe and A. Ohuchi, (2004) DNA solution of the shortest path problem by concentration control, Lecture Notes in Computer Science, vol.2340, pp.203-212, 2004.
8. J. Y. Lee, S. Y. Shin, T. H. Park and B.-T. Zhang, (2004) Solving traveling salesman problems with DNA molecules encoding numerical values, Biosystems, vol.78, no.1-2, pp.39-47, 2004.
9. R. J. Lipton, (1995) DNA solution of hard computational problems, Sciences, vol.268, no.5210, pp.542-545 1995.
10. M. S. Muhammad, Z. Ibrahim, O. Ono and M. Khalid, (2006) Application of length-based DNA computing for complex scheduling problem, International Journal of Information Technology, vol.12, no.3, pp.100-110, 2006.
11. A. Gehani, T. LaBean and J. Reif, (2004) DNA-based cryptography, Lecture Notes in Computer Science, vol.2950, pp.167-188, 2004.
12. S. V. Kartalopoulos, (2005) DNA-inspired cryptographic method in optical communications, authentication and data mimicking, Proc. of the IEEE on Military Communications Conference, vol.2, pp.774-779, 2005.
13. K. Tanaka, A. Okamoto and I. Saito, (2005) Public-key system using DNA as a one-way function for key distribution, Biosystems, vol.81, no.1, pp.25-29, 2005.
14. G. Cui, L. Qin, Y. Wang and X. Zhang, (2007) Information security technology based on DNA computing, Proc. of the 2007 IEEE International Workshop on Anti-counterfeiting, Security, Identification, Xiamen, China, pp.288-291, 2007.
15. S. A. Tsaftaris, A. K. Katsaggelos, T. N. Pappas and E. T. Papoutsakis, (2004) How can DNA computing be applied to digital signal processing?, Signal Processing Magazine, IEEE, vol.21, no.6, pp.57-61, 2004.
16. J. Watada, S. Kojima, S. Ueda and O. Ono, (2004) DNA computing approach to optimal decision problems, Proc. of the 2004 IEEE International Conference on Fuzzy Systems, vol.3, pp.1579-1584,2004.
17. Y. Tsuboi and O.Ono, (2003) Applicability of DNA computing algorithm solving image recognition in intelligent visual mechanics, Proc. of the 29th IEEE Annual Conference, Industrial Electronics Society, vol.3, pp.3800, 2003.

18. L. M Adleman, P. W. K. Rothemund, S. T. Roweis and E. Winfree, (1999). On applying molecular computation to the data encryption standard, Journal of Computational Biology, vol.6, no.1, pp.53-64, 1999.
19. F. Zhang, B. Liu, W. Liu and J. Xu, (2007) A DNA computing model based on acryditeTM gel technology to solve the timetable problem, IEEE/ICME International Conference on Complex Medical Engineering, pp.1632-1635, 2007.
20. Y. Zhixiang, J. Cui, Y. Yang and Y. Ma, (2006) Job shop scheduling problem based on DNA computing, Journal of Systems Engineering and Electronic, vol.17, no.3 pp.654-659, 2006.
21. D. Boneh, C. Dunworth and R. Lipton, (1995) Breaking DES using a molecular computer, Technical Report, CS-TR-489-95, Princeton University, 1995.
22. G. Xiao, M. Lu, L. Qin and X. Lai, (2006) New field of cryptography: DNA cryptography, Chinese Science Bulletin, vol.51, no.12, pp.1413-1420, 2006.
23. D. J.-F. Jeng, J. Watada, B. We and J.-Y. Wu, (2006) Fuzzy forecasting with DNA computing, Lecture Notes in Computer Science, vol.4287, pp.324-336, 2006.
24. D. J.-F. Jeng, I. Kim and J. Watada, (2007) Bio-soft computing with fixed-length DNA to a group control optimization problem, Soft-Computing, Springer, vol.12, no.13, pp.223-228, 2007.
25. I. Kim, D. J.-F. Jeng and J. Watada, (2006) Redesigning subgroups in a personnel network based on DNA computing, International Journal Innovative, Computing, Information and Control, vol.2, no.4, pp.885-896, 2006.
26. R. B. A. Bakar and J. Watada, (2007) A bio-soft computing approach to re-arrange a flexible manufacturing robot, Lecture Notes in Computer Science, vol.4694, pp.308-315, 2007.
27. R. B. A. Bakar and J. Watada, (2007) Biological computation of optimal task arrangement to multiple robot flexible machining cell, Proc. of the International Conference on Soft-Computing and Human Sciences, Kitakyushu, Japan, pp.61-66, 2007.
28. R. A. B. Bakar, J. Watada and W. Pedrycz, (2008) A proximity approach to DNA based clustering analysis, International Journal Innovative, Computing, Information and Control, vol.4, no.5, 2008.
29. R. A. B. Bakar, J. Watada and W. Pedrycz, (2008) DNA approach to solve clustering problem based on a mutual order, Biosystems, vol.91, no.1, pp.1-12, 2008.
30. A.Boukerche, K. R. L. Juca, J. B. Sobral and M. S. M. A. Notare, (2004) An artificial immune based intrusion detection model for computer and telecommunication systems, Parallel Computing, vol.30, no.5-6, pp.629-646, 2004.
31. J. C. Adams, (2008) On the application of DNA based computing, available online at: http ://publish.uwo.ca/ jadams/dnaapps1.htm, Last reviewed 14th January, 2008.
32. Molecular Computing: An Overview by Byoung-Tak Zhang A few ideas about DNA computing: Enrique Blanco
33. Gheorghe Paun, Rozenberg and Salomaa, *DNA computing, New computing paradigms*, Springer-Verlag,Berlin.
34. Gupta, GauraV, Mehra, Nipun and ChakraVerty, *DNA Computing*,
35. Leonard M. Adleman, *Computing with DNA*, Scientific American, August 1998.
36. L.M Adleman, (1994) “Molecular computation of solutions to combinatorial problems,” Science, 266, 1994, 1021-1024.
37. V. Shekhar, Ch.V.Ramana Murthy, P.V. Gopalacharyulu and G.P.Rajasekhar, DNA solution to the shortest path problem Pamela Peters, Ph.D., Access Excellence, Genentech, Inc.