Abstract

DNA (Deoxyribose Nucleic Acid) computing, also known as molecular computing is a new approach to massively parallel computation based on groundbreaking work by Adleman. DNA computing was proposed as a means of solving a class of intractable computational problems in which the computing time can grow exponentially with problem size (the 'NP-complete' or non-deterministic polynomial time complete problems). A DNA computer is basically a collection of specially selected DNA strands whose combinations will result in the solution to some problem, depending on the problem at hand. Technology is currently available both to select the initial strands and to filter the final solution. DNA computing is a new computational paradigm that employs (bio)molecular manipulation to solve computational problems, at the same time exploring natural processes as computational models. In 1994, Leonard Adleman at the Laboratory of Molecular Science, Department of Computer Science, University of Southern California surprised the scientific community by using the tools of molecular biology to solve a different computational problem. The main idea was the encoding of data in DNA strands and the use of tools from molecular biology to execute computational operations. Besides the novelty of this approach, molecular computing has the potential to outperform electronic computers. For example, DNA computations may use a billion times less energy than an electronic computer while storing data in a trillion times less space. Moreover, computing with DNA is highly parallel: In principle there could be billions upon trillions of DNA molecules undergoing chemical reactions, that is, performing computations, simultaneously.

L. M. Adleman launched the field of DNA computing with a demonstration in 1994 that strands of DNA could be used to solve the Hamiltonian path problem for a simple graph. He also identified three broad categories of open questions for the field. First, is DNA capable of universal computation? Second, what kinds of algorithms can DNA implement? Third, can the error rates in the manipulations of the DNA be controlled enough to allow for useful computation? In the two years that have followed, theoretical work has shown that DNA is in fact capable of universal computation. Furthermore, algorithms for solving interesting questions, like breaking
the Data Encryption Standard, have been described using currently available technology and methods. Finally, a few algorithms have been proposed to handle some of the apparently crippling error rates in a few of the common processes used to manipulate DNA. It is thus unlikely that DNA computation is doomed to be only a passing curiosity. However, much work remains to be done on the containment and correction of errors. It is far from clear if the problems in the error rates can be solved sufficiently to ever allow for general-purpose computation that will challenge the more popular substrates for computation. Unfortunately, biological demonstrations of the theoretical results have been sadly lacking. To date, only the simplest of computations have been carried out in DNA. To make significant progress, the field will require both the assessment of the practicality of the different manipulations of DNA and the implementation of algorithms for realistic problems. Theoreticians, in collaboration with experimentalists, can contribute to this research program by settling on a small set of practical and efficient models for DNA computation.

The basics of DNA computing have been discussed in this thesis, its beginning and the concepts behind it. It also describes the characteristics and Nature of DNA computing and why DNA computing is needed. With the advantages and disadvantages of DNA computing, a comparison has been presented between DNA computing and Conventional electronic computers. It has also highlighted the difficulties of DNA computing. We have described the novel Generalised Model, proposed as of our research contribution. The Generalized model has been developed to compute the NP complete problems viz. Hamiltonian Path problem, Maximum clique problem, Sub Graph Isomorphism, Maximum Independent Set and 3-Vertex colouring problem. In this model, each of the above mentioned algorithm has the result of permutation as the input in it. The permuted output is then processed according to the NP Complete as stated above. The complexity and viability of the Generalized Model is also explained. The Fragile and Tough models with the respective complexities. The loopholes of DNA computing are also discussed and how DNA computing has an upper hand over traditional computing has also been discussed.