Simulating DNA Computing

Sanjeev Baskiyar*

Auburn University, Auburn, AL 36849
baskiyar@eng.auburn.edu

Abstract. Although DNA (deoxy-ribo nucleic acid) can perform $10^{22}$
computations per second, it is time intensive and complex to set up
input and output of data to and from a biological computer and to filter
the final result. This paper, discusses how to simulate DNA computing
on a digital computer to solve the Hamiltonian path problem using
Adleman’s model. The simulation serves as an educational tool to teach
DNA computing without the elaborate bio-experiments. As an aside, it
also digitally verifies Adleman’s notion of DNA computing to solve the
Hamiltonian path problem. Future work will involve a parallel
implementation of the algorithm and investigation of the possibility of
construction of simple regular VLSI structures to implement the basics
of the model for fixed-sized problems.

Keywords: Parallel Computing, DNA, Hamiltonian path, Simulation,
Educational tool.

1 Introduction

The idea of viewing living cells and molecular complexes as computing components
dates back to the late 1950s, when Feynman [9] described “sub-microscopic
computers.” Recently, Adleman [1] showed that random highly parallel searches
could be performed using DNA-manipulation techniques.

DNA is a nucleic acid that carries the genetic information in cells and is capable of
self-replication. It consists of two long chains of nucleotides twisted into a double
helix and joined by hydrogen bonds between the complementary bases Adenine (A)
and Thymine (T) or Cytosine (C) and Guanine (G). The two linear chains of
nucleotides must have complementary base sequences and opposite chemical
polarities to form a partial or a full duplex.

Massively parallel computations can be performed by recombinant bio-operations
on many DNA molecules simultaneously. Parallel molecular machines can be
constructed by encoding a problem using DNA strands. The core of a DNA computer
is a collection of selected DNA strands whose combinations result in solutions to
problems. Technology is currently available to select and filter strands.

* Assistant Professor

© Springer-Verlag Berlin Heidelberg 2002
Theoretically, DNA computing has the potential to outperform electronic computers. Adleman's model [2] shows that the number of operations per second ($1.2 \times 10^{18}$) performed by a DNA computer is approximately 12 million times of that of the current fastest supercomputer. Also, DNA computers are energy efficient—a DNA computer [1] performs $2 \times 10^{19}$ operations per joule, whereas the fastest supercomputer executes $10^9$ operations per joule. Information can be stored in DNA molecules at a density of 1 bit/nm$^3$, whereas electronic storage density is approximately 1 bit/10$^{12}$ nm$^3$ [5]. Thus DNA computing may offer excellent improvements in speed, energy consumption and information density.

However, the main obstacles to creating a practical DNA computer remain. Such obstacles are encountered in dealing with the complexity of the experimental setup and errors. DNA computing is efficient in solving large intractable problems such as the directed Hamiltonian path problem, the satisfiability problem (SAT) and for deciphering codes. However, for simple Boolean and arithmetic operations, electronic computers outperform DNA computers in speed. Thus problems, whose algorithms can be highly parallelized, may become the domain of DNA computers, whereas others, whose algorithms are inherently sequential, may remain the specialty of electronic computers.

The organization of this paper is as follows. In Section 2, we review previous work in DNA computing. In Section 3, we develop a digital computer simulation of Adelman's DNA computing technique to solve the Hamiltonian path problem. In Section 4, we conclude by discussing the results of the simulation and suggesting directions for future work.

2 Background

Research in DNA computing has incorporated both experimental and theoretical aspects. Theoretically, it has been shown that using DNA one can construct a Turing machine [12]. Also, mathematical models of the computing process have been developed and their in-vitro feasibility examined. On the other hand, experiments have been performed to solve various problems such as: the satisfiability problem [19], breaking the data encryption standard [7], expansions of symbolic determinants [18], matrix multiplication [22], graph connectivity and knapsack problems using dynamic programming [4], the road coloring problem [13], exascale computer algebra problems [26], and simple Horn clause computation [17]. Other studies include: Kaplan's [16] replication of Adleman's experiment, the partial progress of a Wisconsin team of computer scientists and biochemists in solving a 5-variable instance of the SAT problem using a surface-based approach [21] and addition using horizontal chain-reaction [10].

Various aspects of the ability to implement DNA computing have been experimentally investigated. Notable studies are: the impact of encoding techniques on simplifying Adleman's experiment [8], the complications in using the Polymerase Chain Reaction (PCR) [15], the use of self-assembly of DNA [27], the experimental gap in the design and assembly of unusual DNA structures [25], joining and rotating data with molecules [3], evaluating simple Boolean formulas [11] and the use of ligation [14].