A Simulation Software for DNA Computing Algorithms Implementation

M. S. Muhammad, S. M. W. Masra, K. Kipli, and N. Zamhari

***Abstract*—**The capturing of gel electrophoresis image represents the output of a DNA computing algorithm. Before this image is being captured, DNA computing involves parallel overlap assembly (POA) and polymerase chain reaction (PCR) that is the main of this computing algorithm. However, the design of the DNA oligonucleotides to represent a problem is quite complicated and is prone to errors. In order to reduce these errors during the design stage before the actual in-vitro experiment is carried out; a simulation software capable of simulating the POA and PCR processes is developed. This simulation software capability is unlimited where problem of any size and complexity can be simulated, thus saving cost due to possible errors during the design process. Information regarding the DNA sequence during the computing process as well as the computing output can be extracted at the same time using the simulation software.

***Keywords*—** DNA computing, PCR, POA, simulation software

I. Introduction

EVER since Leonard M. Adleman [1] demonstrated the ability of using molecules of Deoxyribonucleic Acid or DNA as a medium for computation to solve a directed Hamiltonian Path Problem (HPP), the interest in applying DNA to solve similar computational problems have increased tremendously [2, 3, 4]. An in vitro experimental work based on DNA computing approach to solve an engineering scheduling problem in the case of an elevator travel path optimization for a typical building of N floors with M elevators has been presented [5].

One of the main problems during the design process of DNA computing is the synthesizing of DNA oligonucleotides to represent both the input and output of the problem. A mechanism for implementing the DNA computing approach for a much larger and complex problem is needed. The main aim of this research is therefore to develop a simulation software capable of simulating the DNA computing process that can verify the expected result before the actual in vitro experiment is carried out. Since the complexity and costs of the DNA oligonucleotides increases for larger and complex problem, this simulation software will provide a helpful guide for the DNA computing implementation as to eliminate errors during the design process. Information regarding the DNA oligonucleotides sequences for both the input and output could also be extracted from the simulation programme. Before the simulation software is presented, an overview of the elevator scheduling problem and its DNA computing solution design is first explained.

M. S. Muhammad is with the Department of Electronics Engineering, Universiti Malaysia Sarawak, MALAYSIA (phone: 082-583356; fax: 082-583410; e-mail: msaufee@ feng.unimas.my).

S. M. W. Masra, K. Kipli and N. Zamhari are with the Department of Electronics Engineering, Universiti Malaysia Sarawak, MALAYSIA, (e-mail: wmmasnia@feng.unimas.my, kkuryati@feng.unimas.my and znurdiani@feng.unimas.my).

II. Overview Of An Elevator Scheduling Problem

The elevator positions at an instance of a time for a 6 floors building with 2 elevators can be illustrated as in Table I. If the position of each elevator at floor 1, 2, 3, 4, 5, and 6 are represented as nodes V_1, V_2, V_3, V_4, V_5, and V_6 respectively, each of the elevator travel path combinations can thus be represented as a weighted graph. At the same time, the graph edges thus represent the elevator's travel path between floors.

TABLE I
Elevator Positions at an Instance of a Time

Floor No	Elevator A	Elevator B	Hall Call
6		(3 , 2)	
5			
4			↑
3			↓
2			
1	(3 , 5)		

Each of the graph edge weights can thus be mathematically formulated as directly proportional to the elevators travelling time between any floor using

$$\omega_{|j-i|} = (|j - i|)\, T_C + T_S \qquad (1)$$

where

i = elevator's present floor position
j = elevator's destination floor position
$|j - i|$ = total number of floors of elevator's movement
T_C = elevator's travelling time between two consecutive floors
T_S = elevator's stopping time at a floor