Application of Length-Based DNA Computing for an Elevator Scheduling Problem

Mohd Saufee Muhammad, Zuwairie Ibrahim, Osamu Ono, and Marzuki Khalid

Institute of Applied DNA Computing, Meiji University,
1-1-1 Higashi-Mita, Tama-Ku, Kawasaki-Shi, Kanagawa-Ken, 214-8571 Japan
{msaufee, zuwairie, ono}@isc.meiji.ac.jp, marzuki@utmkl.utm.my
http://www.isc.meiji.ac.jp/~i3erabc/IADC.html

Abstract

This paper discussed the implementation ideas and experimental procedures to solve an engineering related combinatorial problem using DNA computing approach. An elevator scheduling problem is chosen as a benchmark where all the elevator travel paths are represented by DNA sequences of specific lengths that represent the elevators traveling time in a proportional way based on certain initial conditions such as elevator's present and destination floors, and hall calls for an elevator from a floor. Parallel overlap assembly method is employed for initial pool generation for a more efficient generation of all possible elevator travel paths. The initial pool generation solution is then subjected to polymerase chain reaction and gel electrophoresis to extract the required optimal path. Experimental result obtained shows that DNA computing approach can be well-suited for solving such real-world application in the near future.

Keywords: DNA computing, elevator scheduling problem, parallel overlap assembly, polymerase chain reaction, gel electrophoresis, optimal path.

1 Introduction

In 1994, Adleman [1] demonstrated the practical possibility of using molecules of Deoxyribonucleic Acid or DNA as a medium for computation. In his experiment, Adleman successfully solved a directed Hamiltonian Path Problem (HPP) using the tools of biomolecular engineering. Adleman [2] created DNA strands to represent an airplane flight from each of the seven cities, and then combined them to produce every possible route. Given its vast parallelism, the DNA strands yielded 10^9 answers in less than one second.

DNA computation relies on devising algorithms that solve problems using the encoded information in the sequence of nucleotides that make up DNA's double helix – the bases Adenine, Guanine, Thymine, and Cytosine (A, G, T, and C, respectively) and then breaking and making new bonds between them to reach the answer.

Research on DNA application to solve engineering problem however has not been very well established. In this paper DNA computing technique to solve such problems is proposed. Since DNA computing is very suitable to solve combinatorial problems, an elevator scheduling problem is chosen to be solved using this computing technique.

The scheduling problem involves finding an optimal path, or in other words, finding the shortest elevator travel paths of a building with certain number of elevators and floors. However, this problem is a complex combinatorial problem since certain criteria need to be fulfilled for the problem solution such as initial elevator position, its destinations and hall calls made for an elevator.

There are several research reports on DNA computing techniques for solving shortest path problems. Among others, a constant proportional length-based DNA computing technique for traveling salesman problem (TSP) has been proposed by Nayaranan and Zorbalas [3]. Yamamoto *et al.* [4] proposed a concentration-controlled DNA computing to accomplish local search for solving shortest path problem. Lee *et al.* [5] proposed a DNA computing technique based on temperature gradient to solve the TSP problem. Ibrahim *et al.* [6] on the other hand proposed a direct-proportional length-based DNA computing for shortest path problem. In this paper, the feasibility of the method proposed by [6] is been tested to find the solution for the optimal path of the elevator scheduling problem. Constraints such as node position in the graph, DNA sequence design and initial pool generation are investigated and discussed in detail for the successful implementation of the DNA computing method used.

2 Elevator Scheduling Problem

Table 1 illustrates an elevator situation at an instance of a time for a building with N floors and M elevators. The elevator travel paths can be represented as a graph by representing the elevator position at floor 1, 2, 3, … , $N-2$, $N-1$, N with nodes V_1, V_2, V_3, … , V_{N-2}, V_{N-1}, V_N respectively. The graph of all possible travel paths of one of the elevators is constructed as shown in Fig. 1.

Table 1. Elevator situation at an instance of time

Floor No	Elevator 1	Elevator 2	…	Elevator $M-1$	Elevator M	Hall Call
N			…	($N-3$, 7, 3)		
$N-1$	($N-2$, 4, 1)		…			↑
$N-2$			…			↓
:	:	:	:	:	:	:
3		(4, 6, $N-2$)	…			↑
2			…		(5, 8, $N-1$)	↓
1			…			

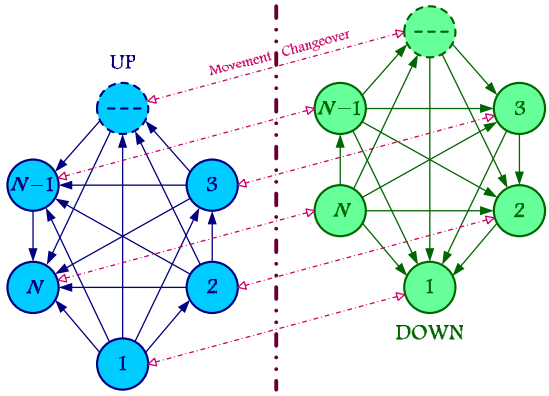

Fig. 1. Graph of all possible travel paths of an elevator

The weight between each node of the graph can be represented as

$$\omega_{|j-i|} = (|\ j-i\ |)T_T + T_S \tag{1}$$

where

i – elevator present floor position
j – elevator destination floor position
$|j-i|$ – total no. of floors of elevator movement
T_T – elevator traveling time between two consecutive floors
T_S – elevator stopping time at a floor

The output of the graph, given by sum of the graph weights thus represents the total traveling time of the elevator, i.e.

$$G(E) = \sum_{|j-i|=1}^{N-1} \omega_{|j-i|} \tag{2}$$

For a building with M elevators, M similar graphs as shown in Fig. 1 can be duplicated representing all M elevator travel paths. The total traveling time of all the elevators can thus be calculated by summing up each of the elevator traveling time as

$$G(E_1, E_2, \cdots, E_{M-1}, E_M) = G(E_1) + G(E_2) + \cdots + G(E_{M-1}) + G(E_M) \tag{3}$$

The optimal travel path is thus given by the minimum total traveling time of all the elevators with all initial conditions and requirements satisfied, i.e.

$$\text{Optimal Travel Path} = G(E_1, E_2, \ldots, E_{M-1}, E_M)_{min} \tag{4}$$

Let us now consider a building with 2 elevators and 6 floors. Elevator A is presently at 1st floor and its destination are 3rd and 5th floors, while elevator B is presently at 6th floor and its destination are 3rd and 2nd floors. There are hall calls at 4th floor going up, and hall calls at 3rd floor going down, as illustrated in Table 2.

Table 2. Elevator position for elevator scheduling problem example

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

If we represent the elevator position at a floor as nodes V_1, V_2, V_3, V_4, V_5 and V_6 representing all the 6 floor positions in the building respectively, all the possible elevator travel paths can be represented as shown in Fig. 2.

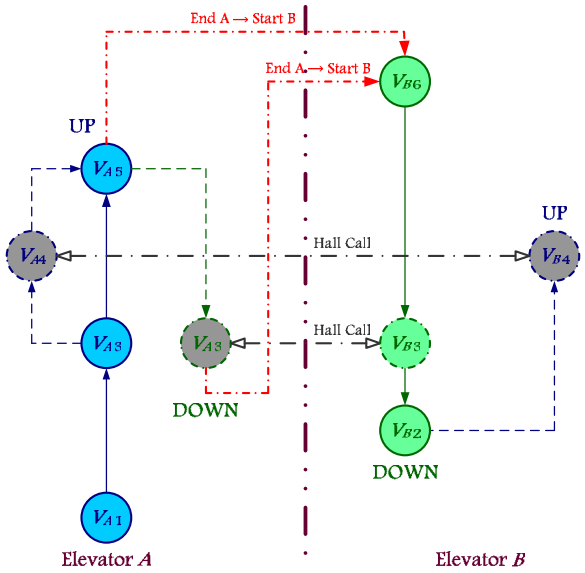

Fig. 2. Graph of all possible travel path combinations of elevators *A* and *B*

Since the building is 6 floors high, the maximum number of floors that the elevator can travel is (6 – 1) = 5 floors. Now, assume that the traveling time between two consecutive floors T_T = 5 sec and the stopping time of the elevator at a floor T_S = 15 sec, and representing 5 sec of time with 10 units we have using (1)

$\omega_1 = 1(5) + 15 = 20$ sec $= 40$, $\omega_2 = 2(5) + 15 = 25$ sec $= 50$

$\omega_3 = 3(5) + 15 = 30$ sec $= 60$, $\omega_4 = 4(5) + 15 = 35$ sec $= 70$

$\omega_5 = 5(5) + 15 = 40$ sec $= 80$

All possible travel paths of elevators *A* and *B* shown in Fig. 2 can now be replaced with a weighted graph as depicted in Fig. 3.

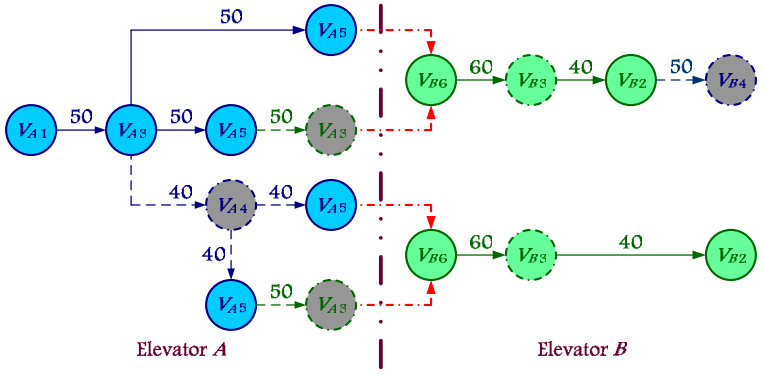

Fig. 3. Weighted graph of all possible travel path combinations of elevators *A* and *B*

The graph output for all possible travel paths can now be calculated. Here, it is clearly seen that there are four possible travel paths with either elevator *A* or *B* answer the hall calls. All the graph outputs can be now be calculated as below

$$G(A, B)_1 = (V_{A1} \rightarrow V_{A3} \rightarrow V_{A5}) + (V_{B6} \rightarrow V_{B3} \rightarrow V_{B2} \rightarrow V_{B4})$$
$$= (50+50) + (60+40+50) = 250 = 125 \text{ sec}$$

$$G(A, B)_2 = (V_{A1} \rightarrow V_{A3} \rightarrow V_{A5} \rightarrow V_{A3}) + + (V_{B6} \rightarrow V_{B3} \rightarrow V_{B2} \rightarrow V_{B4})$$
$$= (50+50+50) + (60+40+50) = 300 = 150 \text{ sec}$$

$$G(A, B)_3 = (V_{A1} \rightarrow V_{A3} \rightarrow V_{A4} \rightarrow V_{A5}) + (V_{B6} \rightarrow V_{B3} \rightarrow V_{B2})$$
$$= (50+40+40) + (60+40) = 230 = 115 \text{ sec}$$

$$G(A, B)_4 = (V_{A1} \rightarrow V_{A3} \rightarrow V_{A4} \rightarrow V_{A5} \rightarrow V_{A3}) + (V_{B6} \rightarrow V_{B3} \rightarrow V_{B2})$$
$$= (50+40+40+50) + (60+40) = 280 = 140 \text{ sec}$$

The minimum output of the graph $G(A, B)_3 = 230 = 115$ sec is thus the required optimal travel path for the elevator scheduling problem stated above.

3 Length-based DNA Computing for Scheduling Problem

In order to solve the elevator scheduling problem for the example above using length-based DNA computing, we first represent the problem as a weighted graph as shown in Fig. 3. The graph is then redrawn in order to distinguish between start, immediate and end nodes and also to differentiate between the different travel path nodes as shown in Fig. 4.

Each of the nodes is then assigned with a unique DNA sequence where each intermediate node of different travel paths is assigned with a specific DNA sequence and each start or end node of different travel paths is assigned with another specific DNA sequence. Hence, every DNA sequence assigned to each node will identify its locations and also its travel paths.

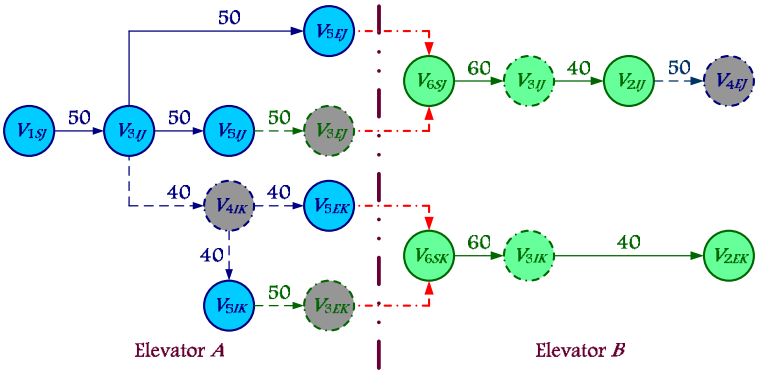

Fig. 4. Weighted graph of all possible travel path combinations of elevators *A* and *B* showing different node locations and paths

The DNA sequences are designed using available software for DNA sequence design named DNASequenceGenerator [7] and is shown in Table 3. The GC contents (GC %), melting temperature (T_m) and the complement of each sequence is also shown in the table where *S*, *I* and *E* denotes start, intermediate and end nodes respectively, while *J* and *K* denotes the different elevator travel paths.

Table 3. DNA sequence for nodes

Node *Vi*	20-mer Sequence (5'–3')	GC%	T_m (°C)	Node $\overline{V_i}$	20-mer Sequence (5'–3')
V_{1SJ}	cggcggtccactaaatacta	50	60.0	$\overline{V_{1SJ}}$	tagtatttagtggaccgccg
V_{3IJ}	cactctttgtgaacgccttc	50	60.8	$\overline{V_{3IJ}}$	gaaggcgttcacaaagagtg
V_{4IK}	gtgggttagaggtagtccgg	60	60.8	$\overline{V_{4IK}}$	ccggactacctctaacccac
V_{5IJ}	tgaaccggccctttatatct	45	60.7	$\overline{V_{5IJ}}$	agatataaagggccggttca
V_{5IK}	ccgctgatccttgctaagta	50	60.4	$\overline{V_{5IK}}$	tacttagcaaggatcagcgg
V_{3EJ}	tcattcgagttattcctggg	45	59.9	$\overline{V_{3EJ}}$	cccaggaataactcgaatga
V_{3EK}	aaatgacctttttaacggca	35	59.4	$\overline{V_{3EK}}$	tgccgttaaaaaggtcattt
V_{5EJ}	ctataaggccaaagcagtcg	50	59.9	$\overline{V_{5EJ}}$	cgactgctttggccttatag
V_{5EK}	atgcctggctaaagtgagac	50	59.3	$\overline{V_{5EK}}$	gtctcactttagccaggcat
V_{6SJ}	ggacctgcatcataccagtt	50	59.8	$\overline{V_{6SJ}}$	aactggtatgatgcaggtcc
V_{6SK}	tgcacgcaaaactatttcat	35	59.2	$\overline{V_{6SK}}$	atgaaatagttttgcgtgca
V_{2IJ}	aaagcccgtcggttaagtta	45	60.8	$\overline{V_{2IJ}}$	taacttaaccgacgggcttt
V_{3IK}	tctgcactgttaatgagcca	45	60.4	$\overline{V_{3IK}}$	tggctcattaacagtgcaga
V_{2EK}	ctacggataggtgtctggga	55	59.9	$\overline{V_{2EK}}$	tcccagacacctatccgtag
V_{4EJ}	ggaatccattgatcgcttta	40	59.9	$\overline{V_{4EJ}}$	taaagcgatcaatggattcc

The oligos for each node path of the graph is then synthesized according to the following rules [11] so that the oligos length will directly represent the weights between the nodes :

(i) If *i* is a start node and *j* is an intermediate node, synthesize the oligo as
$$V_i(20) + W_{ij}(\omega_{ij} - 30) + V_j(20)$$
(ii) If *i* is an intermediate node and *j* is an end node, synthesize the oligo as
$$V_i(20) + W_{ij}(\omega_{ij} - 30) + V_j(20)$$
(iii) If *i* and *j* are both intermediate nodes, synthesize the oligo as
$$V_i(20) + W_{ij}(\omega_{ij} - 20) + V_j(20)$$

where *V* denotes the DNA sequence for node, *W* denotes the DNA sequence for weight, ω denotes the weight value, and '+' denotes a 'join' between the DNA sequence. All the synthesized oligos based on the stated rules are shown in Table 4 where regular alphabet denotes the node and italic alphabet denotes the weight between the nodes.

Table 4. DNA sequence for node paths

Node Path	DNA Sequence (5' – 3')
$V_{1SJ} \rightarrow V_{3IJ}$	cggcggtccactaaatacta*aggtcgtttaaggaagtacg*cactctttgtgaacgccttc
$V_{3IJ} \rightarrow V_{4IK}$	cactctttgtgaacgccttc*acgtcgtgtaacgaagtcct*gtgggttagaggtagtccgg
$V_{3IJ} \rightarrow V_{5IJ}$	cactctttgtgaacgccttc*ccgtcggttaagcaagtaatgtactatgct*tgaaccggccctttatatct
$V_{3IJ} \rightarrow V_{5EJ}$	cactctttgtgaacgccttc*gcgtcgcttaccgaagcacg*ctataaggccaaagcagtcg
$V_{4IK} \rightarrow V_{5IK}$	gtgggttagaggtagtccgg*cgctcgttgaagccagtacc*ccgctgatccttgctaagta
$V_{4IK} \rightarrow V_{5EK}$	gtgggttagaggtagtccgg*gcgtcttttta*tgcctggctaaagtgagac
$V_{5IJ} \rightarrow V_{3EJ}$	tgaaccggccctttatatct*acgtgttttacccaagtcag*tcattcgagttattcctggg
$V_{5IK} \rightarrow V_{3EK}$	ccgctgatccttgctaagta*gcggcgtgtcacgaactac*gaaatgacctttttaacggca
$V_{3EJ} \rightarrow V_{6SJ}$	tcattcgagttattcctgggggacctgcatcataccagtt
$V_{5EJ} \rightarrow V_{6SJ}$	ctataaggccaaagcagtcgggacctgcatcataccagtt
$V_{3EK} \rightarrow V_{6SK}$	aaatgacctttttaacggcatgcacgcaaaactatttcat
$V_{5EK} \rightarrow V_{6SK}$	atgcctggctaaagtgagactgcacgcaaaactatttcat
$V_{6SJ} \rightarrow V_{3IJ}$	ggacctgcatcataccagtt*acgtggtttaaggaagtacggtactatgct*cactctttgtgaacgccttc
$V_{6SK} \rightarrow V_{3IK}$	tgcacgcaaaactatttcat*ccgtgggttaaagaagtcctgtactctcctt*ctgcactgttaatgagcca
$V_{3IJ} \rightarrow V_{2IJ}$	cactctttgtgaacgccttc*acgtcgctgcaagaactacg*aaagcccgtcggttaagtta
$V_{3IK} \rightarrow V_{2EK}$	tctgcactgttaatgagcca*acgtcttgtc*ctacggataggtgtctggga
$V_{2IJ} \rightarrow V_{4EJ}$	aaagcccgtcggttaagtta*ggtcttttaatcaactaatg*ggaatccattgatcgcttta

All the synthesized oligos are then poured into a test tube for initial pool generation. Parallel overlap assembly (POA) [8] is used to for the initial pool generation as suggested by Lee *et al*. [9] who demonstrated that POA is a more efficient and economical method for weighted graph problems. POA operation is similar to polymerase chain reaction (PCR) [10], only that POA operates without the use of primers. As PCR, one cycle consists of three steps: hybridization, extension, and

denaturation. During the annealing step, the temperature is decreased slowly so that partial hybridization is allowed to occur at respective locations. The extension on the other hand is applied with the presence of polymerase enzyme and the polymerization can be done from 5' to 3' direction. The generated double stranded DNA molecules are then separated during denaturation step. This can be done by increasing the temperature until the double stranded DNA molecules are separated to become single stranded DNA molecules. An example of the POA showing the optimal path for this elevator scheduling problem is depicted in Fig. 5.

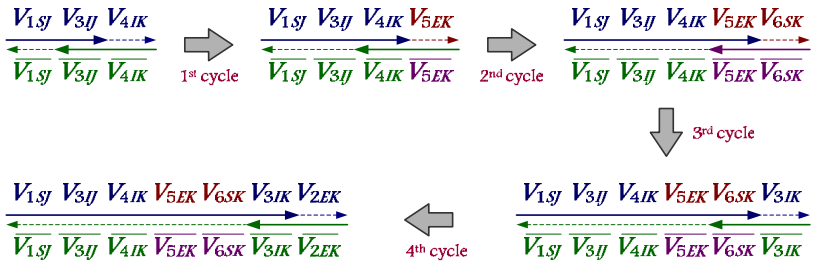

Fig. 5. POA for elevator optimal path representing $V_{1SJ} \rightarrow V_{3IJ} \rightarrow V_{4IK} \rightarrow V_{5EK} \rightarrow V_{6SK} \rightarrow V_{3IK} \rightarrow V_{2EK}$. The continuous arrows represent the synthesized oligos and dotted arrows represent the elongated part during polymerization. The arrowhead indicates the 3' end

At this stage, an initial pool of solution is produced. The optimal path combinations among many other alternative path combinations of the problem have to be filtered. This filtering process copies the target DNA duplex exponentially using the PCR process by amplifying all the DNA molecules containing start node V_{1SJ} and end node V_{2EK}. Numerous amount of DNA strands representing the start node V_{1SJ} and end node V_{2EK} passing through all possible paths will be presented once the PCR operation is accomplished. Finally, gel electrophoresis [11, 12] is then performed onto the output solution of the PCR. The DNA molecules will be separated according to its length during this operation. The bands of gel electrophoresis are then analyzed, and the DNA duplex representing the shortest path starting from node V_{1SJ} and ending at node V_{2EK} will be extracted to represent the required solution of the problem.

4 Experimental Setup and Result

The POA method for initial pool generation is performed in a 100 μl solution consisting of 64.0 μl distilled water (Maxim Biotech), 15.5μl oligos (Proligo Primers & Probes, USA), 10 μl dNTP (TOYOBO, Japan), 10 μl 10× KOD dash buffer (TOYOBO, Japan), and 0.5 μl KOD dash polymerase (TOYOBO, Japan). The solution is then subjected to POA reaction of 25 cycles where the different temperatures for each cycle are 94ºC for 30sec, 55ºC for 30sec and 74ºC for 10sec respectively.

Polymerase chain reaction (PCR) is then performed for DNA amplification in order to select the paths that begin with node V_{1SJ} and ending at node V_{2EK} and V_{4EJ}. PCR is performed in a 25 μl solution consisting of 17.375 μl distilled water (Maxim Biotech), primers V_{1SJ} , $\overline{V_{2EK}}$, and $\overline{V_{4EJ}}$ of 0.5 μl each, 1 μl POA template, 2.5 μl dNTP (TOYOBO, Japan), 2.5 μl 10× KOD dash buffer (TOYOBO, Japan), and 0.125 μl KOD dash polymerase (TOYOBO, Japan). The solution is then subjected to PCR reaction of 25 cycles where the different temperatures for each cycle are 94ºC for 30sec, 55ºC for 30sec and 74ºC for 10sec respectively, i.e. the same as POA process.

Finally, the PCR solution is subjected to gel electrophoresis for 30 minutes in order to visualize the computation result. SYBR Gold (Molecular Probes) is used to stain the gel after gel electrophoresis process before the gel image is captured.

The captured image for the POA and PCR process is shown in Fig. 6. Lane *M* denotes 20bp ladder while lanes 1 and 2 denote POA and PCR product respectively. It is clearly seen from the POA gel image that the band is blurs denoting that all possible travel paths are successfully generated. The PCR gel image shows 4 bands indicating all the four possible travel paths, i.e. $G(A, B)_3$ = 230bp, $G(A, B)_1$ = 250bp, $G(A, B)_4$ = 280bp and $G(A, B)_2$ = 300bp. This confirms the expected result that the optimal elevator's travel path is given by $G(A, B)_3$ = 230bp = 115 sec.

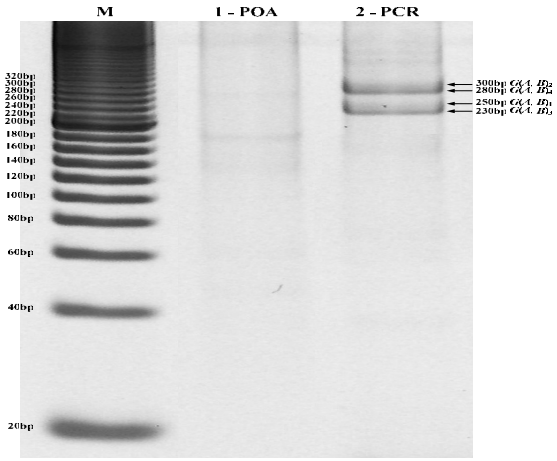

Fig. 6. Experimental results of gel electrophoresis on 10% polyacrylamide gel. Lane *M* denotes 20bp ladder and lanes 1 and 2 is the product of POA and PCR respectively

5 Conclusions

Ideas and implementation procedures for application of length-based DNA computing to solve a complex elevator scheduling problem have been presented and discussed in

details in this paper. This type of engineering problem had been shown to be achievable and applicable to be solved using DNA computing approach. Experimental result that had been carried out verifies that the shortest DNA sequence length represents the required optimal path for the elevator scheduling problem. With this successful result, the applicability of DNA computing could hence be extended into many more complex problems of this type of nature.

References

1. Adleman, L.M.: *Molecular Computation of Solutions to Combinatorial Problems*. Science, Vol. 266 (1994) 1021-1024
2. Adleman, L.M.: *Computing with DNA*. Scientific American (1998) 34-41
3. Narayanan, A., Zorbalas, S.: *DNA Algorithms for Computing Shortest Paths*. Proceedings of Genetic Programming, (1998) 718-723
4. Yamamoto, Y., Kameda, A., Matsuura, N., Shiba, T., Kawazoe, Y., Ahochi, A.: *Local Search by Concentration-Controlled DNA Computing*. International Journal of Computational Intelligence and Applications, Vol. 2 (2002) 447-455
5. Lee, J.Y., Shin, S.Y., Augh, S.J., Park, T.H., Zhang, B.T.: *Temperature Gradient-Based DNA Computing for Graph Problems with Weighted Edges*. Lecture Notes in Computer Science, Springer-Verlag, Vol. 2568 (2003) 73-84
6. Ibrahim, Z., Tsuboi, Y., Ono, O., Khalid, M.: *Direct-Proportional Length-Based DNA Computing for Shortest Path Problem*. International Journal of Computer Science and Applications, Vol. 1, Issue 1 (2004) 46-60
7. Udo, F., Sam, S., Wolfgang, B., Hilmar, R.: *DNASequenceGenerator: A Program for the Construction of DNA Sequences*. Proceedings of the Seventh International Workshop on DNA Based Computers (2001) 23-32
8. Kaplan, P.D., Ouyang, Q., Thaler, D.S., Libchaber, A.: *Parallel Overlap Assembly for the Construction of Computational DNA Libraries*. Journal of Theoretical Biology, Vol. 188, Issue 3 (1997) 333-341
9. Lee, J.Y., Lim, H.W., Yoo, S.I., Zhang, B.T., Park, T.H.: *Efficient Initial Pool Generation for Weighted Graph Problems Using Parallel Overlap Assembly*. Preliminary Proceeding of the 10th International Meeting on DNA Computing (2004) 357-364
10. Fitch, J. P.: *Engineering Introduction to Biotechnology*. SPIE Press (2001)
11. Paun, G., Rozenberg, G., Salomaa, A.,: *DNA Computing: New Computing Paradigms*. Lecture Notes in Computer Science, Springer-Verlag, Vol. 1644 (1998) 106-118
12. Yamamoto, Y., Kameda, A., Matsuura, N., Shiba, T., Kawazoe, Y., Ahochi, A.: *A Separation Method for DNA Computing Based on Concentration Control*. New Generation Computing, Vol. 20, No. 3 (2002) 251-262

Biography

Mohd Saufee Muhammad received his B.Eng (Electronic Computer Systems) from University of Salford, UK in 1996 and MSc in Engineering (Electrical) from Western Michigan University, USA in 2000. He is a staff member of Department of Electronics, Faculty of Engineering, Universiti Malaysia Sarawak, Malaysia since 2000. He is currently on a study leave pursuing his PhD at the Institute of Applied DNA Computing, Meiji University, Kanagawa, Japan. He is a student member of Institute of Electrical and Electronics Engineers (IEEE), IEEE Computational Intelligence Society (ICIS), and IEEE Signal Processing Society (ISPC). His research interests include DNA computing applications and artificial intelligence.

Zuwairie Ibrahim received his B.Eng (Mechatronics) and M.Eng. (Image Processing) from Universiti Teknologi Malaysia, Malaysia, in 2000 and 2002 respectively. Since 2002, he is engaged with Department of Mechatronics and Robotics, Universiti Teknologi Malaysia as a lecturer. He is currently pursuing his PhD at the Institute of Applied DNA Computing, Meiji University, Kanagawa, Japan. He is a student member of Institute of Electrical and Electronics Engineers (IEEE), International Computational Intelligence Society (ICIS), International Society for Nanoscale Science, Computation and Engineering (ISNSCE), and International Signal Processing Society (ISPC). His research interests include signal and image processing, automated visual inspection, evolutionary and unconventional computing such as molecular or DNA computing.

Osamu Ono received his Bachelor, Master and Doctor Degree in Engineering from Waseda University, Tokyo, in 1974, 1976, and 1979 respectively. He is a Professor of the Department of Electrical and Electronic Engineering, Meiji University, Japan. He is also the Director of Tokyo Branch, Japan Institute Electrical Engineering and a committee member of Japan Society for Simulation Technology (JSST). His research interest includes large scale industrial process, mechatronics, advanced mobile robotics and image processing. Presently, his research interest is mainly on the applications of DNA computing in engineering field.

Marzuki Khalid is currently a Professor of Intelligent Control and Director of Centre of Artificial Inteligence and Robotics (CAIRO), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. His current research interest is in the field of artificial intelligence with applications to control. He is a member of Editorial Advisory Board, International Journal of Engineering Applications of Artificial Intelligence published by Elsevier Science and an Associate Editor of the Journal of Systems and Control Engineering published by the Institute of Mechanical Engineers, United Kingdom. He is currently the IEEE Student Activities Chair for Region 10 (Asia Pacific) and also the Founding Member of the Asian Control Professors Association.