Using Fixed Column Primer for Computation of Boolean Matrix Multiplication with DNA

N. Rajaee[1], Tay Kai Meng[1], Asrani Lit[1], Sharifah Masniah Wan Masra [1]

[1]*Faculty of Engineering, University Malaysia Sarawak*

94300 Kota Samarahan, Sarawak, Malaysia

[1] (rnordiana, kmtay, lasrani, wmmasnia)@feng.unimas.my

***Abstract* —** In our previous work, we implemented an in vitro implementation of Boolean matrix multiplication with DNA computing. However, with the increase in the problem size, the material consumption of DNA and the number of experimental steps required to compute the problem increases drastically. Thus in this paper, we introduce fixed column primer method to reduce the material consumption and the labour intensiveness of the computation.

Keyword: DNA Computing, Boolean matrix multiplication, Fixed Column Primer

I. INTRODUCTION

Using DNA for computation has gathered interest in many fields of application due to its massive parallel processing capability. With high density and low energy dispersion, DNA can compute up to 10^{14} maximum operation per second in a single test tube [1]. Such characteristics of DNA computing made it an alternative for solving NP hard problems which require brute force in traditional computing. However, several drawbacks of DNA computing became obvious after its implementation in laboratory works – one of which is the exponential growth of DNA volume required to solve a problem according to its problem size.

In our previous work, we represented the matrix multiplication problem as a directed graph problem and synthesize DNA oligonucleotides to represent the vertices and edges in the graph. However, it shows that for a larger N×N Boolean matrix multiplication with DNA, the volume of the DNA increases and the number of experimental work becomes tedious and impractical to be considered as a viable technology [2,3]. Thus, in this paper we propose a strategy to reduce the experimental protocols and the material consumption for solving a larger set of Boolean matrix multiplication with DNA computing by using a fixed primer to represent the whole set of row indicators for the product matrix.

II. BOOLEAN MATRIX MULTIPLICATION

Boolean matrix multiplication is a fundamental operation that is used in many scientific areas of research such as linear algebra, signal processing, digital control and graph theory. Matrix theory over Boolean algebra is also widely investigated because of its numerous applications in discrete dynamical systems, fuzzy set theory and optimization problems. Although Boolean matrix multiplication problem itself is a simple problem, our main motivation to implement matrix multiplication with DNA computing is to introduce an easier design and read-out model which may be used for computing other matrices problem with DNA.

Consider a binary matrix Y with dimension $m \times n$. According to Kim (1972), Matrix $Y=A \cdot B$ can be represented as a product of two matrices A and B with dimensions $m \times k$ and $k \times n$ respectively. Two Boolean matrices and their product can be represented by a graph problem. The row indicators for the first matrix A and the product matrix Y are represented as initial vertices; the column indicators for the second matrix B and the product matrix Y are represented as terminal vertices; and the column indicators for the first matrix A and the row indicators for the second matrix B are represented as intermediate vertices. For all elements of value 1 in the matrices A and B, a directed edge is drawn for the corresponding initial vertex-intermediate vertex or intermediate vertex-terminal vertex intersections. Elements of value 1 in the product matrix Y is determined by the existence of a "path" which is a continuous linkage of directed edges from an initial vertex to a terminal vertex.

III. COMPUTING BOOLEAN MATRICES WITH DNA

In our previous work, we implemented a Boolean matrix multiplication problem with DNA computing. Let two Boolean matrices A and B and its multiplication product Y, be represented by a graph problem G. We implemented the matrix multiplication problem with a 5 step algorithm:

(Step1) Design of sequences for Vertices and Edges in G.
(Step2) Generate an initial pool of possible solutions
(Step3) Determine Primers for Filtering.
(Step 4) Remove unwanted strands/Mass Copying.
(Step 5) Identify strands.

DNA computing is a wet-lab process which includes a number of feasible bio-chemical operations (tools) necessary to execute such computations. Basic bio-chemical tools included in the experiment protocols are briefly discussed in this section:

Parallel Overlap Assembly (POA) method was successfully applied by Kaplan et al. for initial pool generation consisting

of binary numbers to solve a maximal clique problem with DNA computing. The initial pool is a combinatorial library containing numerical or indicative information represented by DNA sequences. Construction of computational DNA libraries is based on a DNA shuffling method consisting of two parts; one is the position string of fixed length and the other is value string (0 or 1) of various lengths. The DNA strands corresponding to the same position string are overlapped during annealing step in the assembly process while the remaining parts of the DNA strands are extended by dNTPs incorporation by polymerase. During each cycle in POA, the DNA strands self assemble and extend/elongate as the denaturation and annealing processes are repeated causing the number of target strands decreasing while the lengths of the newly formed strands increasing [7].

Polymerase Chain Reaction (PCR) is an amplification technique widely used in molecular biology. A pair of DNA sequences known as "primers" is used to signal the starting point and ending point for a specific target DNA sequence for amplification. The PCR process is capable of exponentially amplify a DNA strand into millions of its copies given a site-specific single molecule DNA and the process is usually carried out in three stages of different temperatures [7].

Gel electrophoresis is a technique used for separation of DNA strands according to their sizes using electric current applied to the gel containing the strands. The size of the DNA strands refers to the weight of the DNA strands which is proportional to the lengths of their sequences. This technique is based on the fact that DNA molecules are negatively charged. Since DNA molecules have the same charge per unit length, they all migrate at the same speed in aqueous solution. However, if electrophoresis is carried out in gel, the migration rate is affected by its size causing less weighted strands to migrate faster. Thus, sorting the strands by their sequence lengths is made possible using this technique. The results of gel electrophoresis process can be viewed by staining gel with fluorescent dye and photographed under UV light.

We generate randomly unique single stranded DNA sequences to represent all vertices in graph *G*. The DNA sequences for directed edges are designed to be connector strands from an initial vertex to an intermediate vertex or from an intermediate vertex to a terminal vertex. The primers for filtering are determined from the row and column indicators for the product matrix Y to amplify only "path" strands during the mass copying. The elements in the product matrix are represented by test tubes containing primer combinations for each row and column. The value of the elements in the product matrix is identified from the constructed solution "paths" viewed in the gel electrophoresis process which yield highlighted bands for amplified "path" strands.

Computation of Boolean matrix multiplication with DNA normally requires an $m + n$ number of primers to be used as row and column indicators in the product matrix; and an $m \times n$ number of test tubes to represent all the elements in the product matrix. Thus, for a 10×10 product matrix, a total of 20 primers are used to represent the row and column indicators; and a 100 test tubes are needed to represent all elements in the product matrix.

	a – j			B1 – B10			B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
A1	1 0 0 0 0 0 0 0 0 0		a	0 0 0 0 0 0 0 0 0 1		A1	0	0	0	0	0	0	0	0	0	1
A2	0 0 0 1 0 0 1 0 0 0		b	0 1 0 0 0 0 0 0 0 0		A2	1	0	0	1	0	0	0	0	0	0
A3	0 0 0 0 0 0 0 0 1 0		c	0 0 1 0 0 0 0 0 0 0		A3	0	0	0	0	1	0	0	0	0	0
A4	0 1 0 0 0 0 0 0 0 0		d	1 0 0 0 0 0 0 0 0 0		A4	0	1	0	0	0	0	0	0	0	0
A5	0 0 0 0 0 1 0 0 0 0	X	e	0 0 0 0 0 0 0 0 1 0	=	A5	0	0	0	0	0	0	0	1	0	0
A6	0 0 0 0 0 0 0 1 0 0		f	0 0 0 0 0 0 0 1 0 0		A6	0	0	0	0	0	1	0	0	0	0
A7	0 0 0 0 1 0 0 0 0 0		g	0 0 0 1 0 0 0 0 0 0		A7	0	0	0	0	0	0	0	0	1	0
A8	0 0 1 0 0 0 0 0 0 0		h	0 0 0 0 0 1 0 0 0 0		A8	0	0	1	0	0	0	0	0	0	0
A9	0 0 0 0 0 0 0 0 0 1		i	0 0 0 0 1 0 0 0 0 0		A9	0	0	0	0	0	0	1	0	0	0
A10	0 0 0 1 0 0 0 0 0 0		j	0 0 0 0 0 0 1 0 0 0		A10	1	0	0	0	0	0	0	0	0	0

Fig. 1 Two 10 x 10 matrices and its product

IV. FIXED COLUMN PRIMER

We consider a matrix problem in Figure 1. The main concept of a fixed column primer is to represent the whole set of row indicators with a fixed starting strand sequence. Instead of *10* different primer sequences as row indicators, the whole column for initial vertices is represented by one common primer sequence ($m = 1$). The individual rows are distinguished from each other by manipulating the different lengths for each initial vertex, to allow each element in the product matrix arrive at different path lengths, respective to only the corresponding row.

	(10-mer)		(20-mer)	(20-mer)									
				B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
A1	A	10	a	0	0	0	0	0	0	0	0	0	*60*
A2	A	20	b	*70*	0	0	*70*	0	0	0	0	0	0
A3	A	30	c	0	0	0	0	*80*	0	0	0	0	0
A4	A	40	d	0	*90*	0	0	0	0	0	0	0	0
A5	A	50	e	0	0	0	0	0	0	0	*100*	0	0
A6	A	60	f	0	0	0	0	0	*110*	0	0	0	0
A7	A	70	g	0	0	0	0	0	0	0	0	*120*	0
A8	A	80	h	0	0	*130*	0	0	0	0	0	0	0
A9	A	90	i	0	0	0	0	0	0	*140*	0	0	0
A10	A	100	j	*150*	0	0	0	0	0	0	0	0	0

Fig. 2 Fixed Column Primer for Initial Vertices

Figure 2 shows all strands for the initial vertices sharing a common 10-mer fixed column primer sequence A. Each row indicator is then varied in their lengths to retain their separate row distinctiveness. With a fixed column primer, only an $m + n$ primer and n test tubes are necessary to compute the Boolean matrix multiplication. In this case, primers required for row indicators in Figure 2 is an $m = 1$ and for column indicators is $n = 10$. This drastically reduces the total number of primers to $m + n = 11$. Since the fixed column primer is shared by the row indicators and the detection of path is dependent on the number of primer combinations, elements in the product matrix can be verified with only $1 \times n$ test tubes. Hence, a total of only 10 test tubes are sufficient. This drastically reduces the experimental steps needed to identify the elements in the product matrix.

The elements in the product matrix are verified by DNA sequence lengths in each column. An element in a product matrix is defined by its total length:

Total Length =		*fixed column primer*
(path)	+	*additional sequence for initial vertex*
	+	*intermediate vertex*
	+	*terminal vertex*

V. DESIGN AND SYNTHESIS

We implemented a (3x3) x (3x3) Boolean matrix multiplication as in Figure 3 with a fixed column primer.

$$\begin{array}{c} \\ A10\,(20) \\ A20\,(30) \\ A30\,(40) \end{array} \begin{array}{c} a\ b\ c \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} X\ Y\ Z \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \\ A10 \\ A20 \\ A30 \end{array} \begin{array}{c} X\ Y\ Z \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

(20)(20)(20) (20)(20)(20) 60 b.p 70 b.p 80 b.p

Fig. 3 (3x3) x (3x3) Boolean Matrix Multiplication

A 10-mer DNA sequence "`tccccgttcc`" is embedded in all three initial vertex sequences A10, A20, A30. Additional unique sequences are constructed to distinguish the vertices with varied lengths of 20-mer, 30-mer and 40-mer respectively. Intermediate vertices and terminal vertices are set at 20-mer each.

A*-10** ***TCCCCGTTCC **tattcgccta (20 mer)**
A*-20** ***TCCCCGTTCC **acctcggttaaggaagtacg (30 mer)**
A*-30** ***TCCCCGTTCC **ccctcttttaagcaagtaatgtactatgcg (40 mer)**

Fig. 4 Fixed column primer sequences for Initial Vertices

The DNA sequences for all vertices are shown in Table I and the DNA sequences for the edges are shown in Table II.

TABLE I
DNA SEQUENCES FOR VERTICES WITH FIXED COLUMN

Vertex	DNA Sequence (5' – 3')	Length (mer)
A10	tccccgttcctattcgccta	20
A20	tccccgttccacctcggttaaggaagtacg	30
A30	tccccgttccccctcttttaagcaagtaatgtactatgcg	40
a	tcaagcatcgggtcgcaact	20
b	cctatccacggcttggggtc	20
B1	ccaacgagggtgtattccgc	20
B2	ctcagtgccgaaccttgcct	20
B3	aggacatacagaggcgggca	20

TABLE II
DNA SEQUENCES FOR DIRECTED EDGES

Edges	DNA Sequence (5' – 3')	Length (mer)
A10a	cgatgcttgataggcgaata	20
A20b	cgtggataggcgtacttcct	20
A30a	cgatgcttgacgcatagtac	20
aB1	ccctcgttggagttgcgacc	20
bB2	cggcactgaggaccccaagc	20
bB3	tgtatgtcctgaccccaagc	20

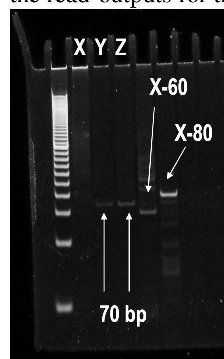
VI. EXPERIMENT

All synthesized DNA sequences for vertices and edges are poured together into a single T0. Parallel Overlap Assembly method is used to generate an initial pool containing all possible solution to the problem. The initial pool generation is performed in a solution containing 67.5µl ddH20 (Maxim Biotech), 1µl for every DNA strand (oligo) (Proligo Primers & Probes), 10µl of dNTP (TOYOBO, Japan), 10µl KOD dash buffer (TOYOBO, Japan) and 0.5µl KOD dash (TOYOBO, Japan). The solution is run for 25 cycles with the first stage of 90ºC for 30 seconds, second stage of 55ºC for 30 seconds and third stage of 74ºC for 10 seconds per cycle. At the end of the POA cycles, the strands for "paths" are formed.

PCR is conducted for all test tubes in Table III, each containing 13.875µl ddH20 (Maxim Biotech), 1µl of template DNA from POA process, 2.5µl of dNTP (TOYOBO, Japan), 2.5µl KOD dash buffer (TOYOBO, Japan) and 0.125µl KOD dash (TOYOBO, Japan) and 2.5 µl of each primer combination. The solutions are spun for 13000 rpm in 25 ºC for 5 minutes before running them for 25 cycles with the first stage of 90ºC for 30 seconds, second stage of 55ºC for 30 seconds and third stage of 74ºC for 10 seconds per cycle.

VII. RESULTS AND CONCLUSION

The experimental results are shown in Figure 4. The lanes for each test tube representing the columns X, Y and Z show the read-outputs for the Boolean matrix computation.



Paths:

A10 → b → X = 60 b.p.
(20) (20) (20)

A20 → a → Y = 70 b.p.
(30) (20) (20)

A20 → a → Z = 70 b.p.
(30) (20) (20)

A30 → c → X = 80 b.p.
(40) (20) (20)

Fig. 4 Results from Gel Electrophoresis

From the results, lanes for column Y and Z yield 70 b.p. highlighted bands which are consistent with our predicted

outcomes. However, the lane for column X is devoid of such highlighted band. We conduct a second PCR to confirm whether the predicted 60 b.p. and 80 b.p. length paths exist for the column X and the results are successful. Both lanes for X-60 and X-80 yield highlighted bands.

The primer A has no problem identifying and amplifying all strands with the defined starting sequences. However, the non-existence of highlighted bands for 60 b.p and 80 b.p in lane X when the paths of 60 b.p and 80 b.p actually exist indicates a problem in Step 4. While the mass amplification works successfully for identification of one type of path strand in a tube (Y and Z), it was less successful in amplifying more than one path in a tube (X).

Another problem with using the fixed column primer algorithm is the restriction of length for initial vertex sequence strands. While in theory, the length of the initial vertex sequence is varied to retain the rows distinctiveness, in actual there is a practical limit to the length of vertex sequence for effective construction of path. Generated DNA sequence strands longer than 80 b.p immediately increase in its melting temperature with some strands having ≥ 100°C. This will cause a problem in the processes employed to implement Step2 and Step4 which have three temperature stages. Vast differences in melting temperatures between the shorter and longer strands will cause the strands with higher melting temperature to not anneal when the shorter strands with lower melting temperature do. Similarly, the shorter strands with lower melting temperature will not denature when the longer strands do which leads to mishybridizations and incomplete paths.

VIII. CONCLUSION

We proposed a fixed column primer to reduce the material consumption and experimental works for computing Boolean matrix multiplication with DNA computing. The results successfully yield predicted outcomes of constructed path but the extraction process is less effective.

ACKNOWLEDGMENT

The first author would like to thank University Malaysia Sarawak.

REFERENCES

[1] R. Deaton, M. Garzon, J. Rose, D.R Franceschetti and S.E Stevens Jr, “DNA Computing: A Review”, Funfamenta Informaticae, IOS Press, Vol 30, pp. 23-41, 1997

[2] N. Rajaee, Y. Kon, K. Yabe and O. Ono, Matrix multiplication with DNA based computing: a comparison study between hybridization-ligation and parallel overlap assembly, Proc. of the 4th IEEE International Conference on Natural Computation, Jinan, China, pp. 527-530, 2008.

[3] K.H Zimmerman, Efficient DNA sticker algorithm for NP-complete graph problems, Computer Physics Communications 144, pp. 297-309, 2002.

[4] L. Adleman, “Molecular computation of solutions to combinatorial problems,” Science, vol. 266, 1994, pp. 1021-1024.

[5] Martyn Amos, Theoretical and Experimental DNA Computation, Natural Computing Series, Springer, 2004

[6] N. Rajaee, H. Aoyagi, K. Yabe, Y. Kon and O.Ono, A new approach for Boolean matrix multiplication with DNA computing, Proc. of IEEE Conference on Soft Computing in Industrial Applications, Hokkaido, Japan, pp. 44-48, 2008.

[7] [16]N. Rajaee, Y. Kon, K. Yabe and O. Ono, Matrix multiplication with DNA based computing: a comparison study between hybridization-ligation and parallel overlap assembly, Proc. of the 4th IEEE International Conference on Natural Computation, Jinan, China, pp. 527-530, 2008.

[8] P. D. Kaplan, D.S. Thaler and A. Libchaber, Parallel overlap assembly of paths through a directed graph, Preliminary Proc. of the 3rd DIMACS Workshop on DNA based computers, Philadelphia PA, pp. 127-141、1997.

[9] .J-F. Jeng, I. Kim and J. Watada, Bio-Inspired evolutionary method for cable trench problem, International Journal of Innovative Computing, Information and Control, vol. 3, no. 1, pp. 111-118, 2007.

[10] Z Ezziane, DNA Computing: applications and challenges, Nanotechnology, Institute of Physics Publishing, vol.17, pp R27-R39, 2006.

[11] John S. Oliver, Matrix multiplication with DNA, J Mol, Evol 45, pp. 161-167, 1997.

[12] R. Agrawal, F. Gustavson and M. Zubair, A high-performance matrix multiplication algorithm on a distributed memory parallel computer using overlapped communication, IBM Journal of Research and Development, vol 38, no. 6, pp 673-681, 1994.

[13] J. Watada, S. Kojima, S. Ueda and O. Ono, DNA computing approach to optimal decision problems, International Journal of Innovative Computing, Information and Control, vol. 2, no. 1, pp. 273-282, 2006.

[14] N. K. Khalid, Z. Ibrahim, T. B. Kurniawan, M. Khalid, N. H. Sarmin and A. P. Engelbrecht, Function minimization in DNA sequence design based on continuous particle swarm optimization, ICIC Express Letters, vol. 3, no. 1, pp. 27-32, 2009.