TRUNCATED BOOLEAN MATRICES FOR DNA COMPUTATION

Nordiana Rajaee[1] , Awang Ahmad Sallehin Awang Hussaini[2] and Sharifah Masniah Wan Masra[4]

[1]Faculty of Engineering, Universiti Malaysia Sarawak
[2]Faculty of Resource Science and Technology, Universiti Malaysia Sarawak
[3]Faculty of Engineering, Universiti Malaysia Sarawak

ABSTRACT

Although DNA computing has emerged as a new computing paradigm with its massive parallel computing capabilities, the large number of DNA required for larger size of computational problems still remain as a stumbling block to its development as practical computing. In this paper, we propose a modification to implement a physical experimentation of two Boolean matrices multiplication problem with DNA computing. The Truncated Matrices reduces the number of DNA sequences and lengths utilized to compute the problem with DNA computing.

KEYWORDS

DNA computation, Boolean Matrices, Bio-molecular tools, Parallel Overlap Assembly

1. INTRODUCTION

When Leonard M Adleman first proposed the use of DNA for computation in solving the Hamiltonian Path Problem (HPP) in 1994, the computation was implemented in an in-vitro experimentation with designed DNA oligonucleotide sequences to represent the vertices and the edges. The solution to the computation was then derived from the chemical reactions via bio-molecular tools such as hybridication-ligation method, polymerase chain reaction and cutting by restriction enzymes. The output was then visualized in gel electrophoresis process. The computation of seven-node HPP took seven days to complete [1]. Since then, many proposals were presented to compute problems with DNA computation but most of them still rely on the L.M Adleman's architecture to carry out the computation. Although the massive parallel computing capabilities of DNA computing promises faster and denser computation there remain several drawbacks which prevent it from becoming a practical computing material. One reason is the exponential requirement of DNA in computing larger size of computational problem [2].

Current strategy in DNA computing is to embed the computation problems in the DNA oligonucleotides sequences and derive the solution by eliminating incorrect DNA via selective processes. For a seven-node HPP, the problem was encoded in a 20 oligonucleotide sequence. For a 23-node HPP, the computation will require 1 kg of DNA and for a 70-node HPP, the computation will require 10^{25} kg of DNA to represent all the nodes [3]. Other problems such as maximal clique problems, vertex-cover problems and set packaging problems all show similarly exponential requirement of DNA and increased time for the computations. LaBean et al (2000) proposed that an1.89^n volume, O (n^2+m^2) time molecular algorithm for the 3-coloring problem

DOI : 10.5121/ijcsea.2014.4501

and a 1.51^n volume, O (n^2m^2) time molecular algorithm for the independent set problem, where n and m are, subsequently, the number of vertices and the number of edges in the problems resolved [4]. Fu (1997) presented a polynomial time algorithm with 1.497^n volume for the 3-SAT problem, a polynomial time algorithm with a 1.345^n volume for the 3-coloring problem and a polynomial time algorithm with a 1.229^n volume for the independent set [5]. Bunow goes on to estimate that an extension combinatorial database would require nearly 10^{70} nucleotides (by comparison, the universe is estimated to contain roughly 10^{80} subatomic particles) [7].

The original algorithm to solve a Boolean matrix multiplication with DNA requires the generation of initial vertices, intermediate vertices, terminal vertices and directed edges to link the initial-intermediate / intermediate-terminal vertices. In our previous works in solving Boolean Matrices with DNA computing, the quantity of DNA oligonucleotides to encode the problem is proportionate to the number of vertices and edges existing in the graph problem representing the matrix multiplication. The number of primers to represent the elements in the product matrix is derived from its total number of row and column indicators whereas the total tubes to represent each element in the product matrix is derived from the total number of primer combinations [7-9]. For a 2 x 2 product matrix, the total number of primers required is 4 and the total number of tubes is also 4. In other words, for an (m x k) • (k x n) matrix multiplication problem, the total number of primers is m + n and total number of tubes is m x n is required to represent the problem.

Given any matrix multiplication problem with increasing N number of intermediate matrices, the number of intermediate vertices for the problem also increases (though not necessarily the number of test tubes representing the product matrix). For example, consider matrix problems with pth power, a $(10 \times 10)^2$ matrix and a $(10 \times 10)^{10}$ matrix multiplication. The number of test tubes representing both problems is 100 but the number of intermediate vertices is 10 (for p = 2) and 90 (for p = 10). The number of primers and tubes increases also drastically for a larger N x N computation. For a 10 x 10 product matrix, the total number of primers required is 20 and the total number of tubes to represent all elements in the product matrix is 100 as shown in Figure 1. As the size of the problem increases, the volume of DNA increases exponentially and the number of experimental work becomes tedious and impractical to be considered as a viable technology.

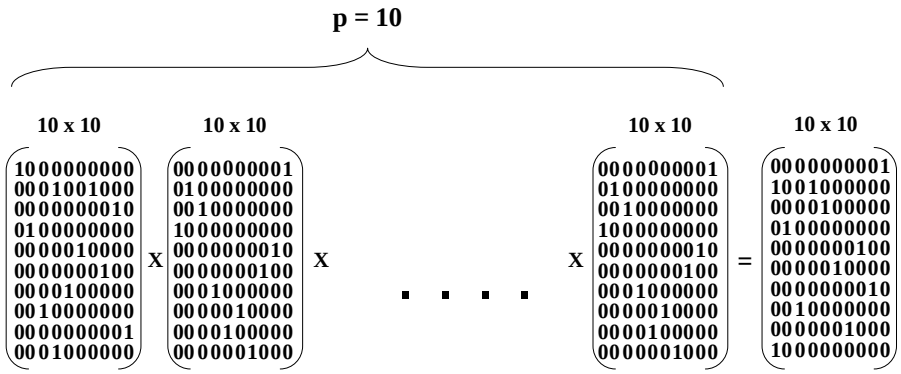

Figure 1. $(10 \times 10)^{10}$ matrix multiplication

2. TRUNCATED BOOLEAN MATRICES

In in-vitro implementation of Boolean Matrix Multiplication problems, DNA oligonucleotide strands are synthesized to represent the initial vertices, intermediate vertices, terminal vertices and edges representing the problem. All generated single stranded sequences are quoted in 5'-3'

order and length is denoted in mer, in which one mer represents one DNA oligonucleotide. The output of the computation is analysed quantitatively based on the direct proportionality of the DNA sequence strands.

Let us assume a Boolean matrix multiplication problem as shown in Figure 2.

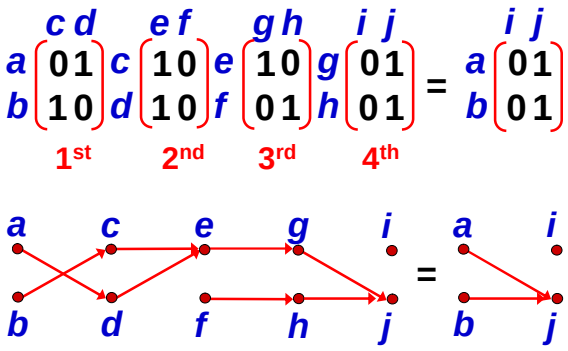

Figure 2. Four Boolean Matrix Multiplication and its graph representation

Using the original algorithm, the problem is represented by a directed graph G where the total DNA strands generated for the problem is 18 as shown in Table 1.

Table 1 Number of DNA sequence strands for vertices and edges

DNA strands sequences for vertices and edges		Number of DNA strands
Initial vertices	$= \{V_a, V_b\}$	2
Intermediate vertices	$= \{ V_c, V_d, V_e, V_f, V_g , V_h\}$	6
Terminal vertices	$= \{V_i, V_j \}$	2
Directed edges	$= \{E_{ad}, E_{bc}, E_{ce}, E_{de}, E_{eg}, E_{fh}, E_{gj}, E_{hj}\}$	8
	Total:	18

In case of cube matrices whereby the number of rows and columns for all matrices in the multiplication are equal, we propose a modification in order to reduce the generation of vertices with Truncated Matrix. The main idea of truncating matrices is to eliminate or reduce the generation of vertices by replacing them directly with directed edges. Originally, the directed edge for element of value 1 in the matrices is constructed from partial complementary strands from a previous vertex to a next vertex. With Truncated Matrices, edges are constructed directly from combination of sequences for rows and columns containing an element value of 1.

Thus, the problem in Figure 2 is modelled such that each row and column indicator for all the matrices refers to a library containing pre-defined sequences for a vertex label. Table 2 shows the library of generated DNA strand sequences for the vertex labels.

Table 2 DNA Sequence for Vertex Labels

	Sequence	Complementary Sequence
a	cgatggcgtg	gctaccgcac
b	cccgagcgtt	gggctcgcaa
c	ggacagccct	cctgtcggga
d	tgccgtagcg	acggcatcgc
e	caacggtggc	gttgccaccg
f	ctctcaggcg	gagagtccgc
g	gctcctgggg	cgaggacccc
h	gagggcgtcg	ctcccgcagc
i	cgatggcgtg	gctaccgcac
j	ggggcggaat	ccccgcctta

The edges constructed to represent the problem are shown in Table 3 and Table 4 for odd and even matrices existing in the problem. SeqLen represents the sequence length of the strands, GC% represent the GC content in the sequence strands which may influence the stability of the strands, and Tm refers to the melting temperature.

For matrix = Odd:

Generate edge sequences for all elements of value 1 from corresponding row and column sequences. From the 1st matrix, elements of value 1 exist for intersections of *row a – column d* and *row b – column c*. From the 3rd matrix, elements of value 1 exist for intersections of *row e – column g* and *row f – column h*. Therefore, the generated edges for the odd matrices are as shown in Table 3:

Table 3 DNA Sequence for Edges (Matrix: Odd)

Edges	Sequence	SeqLen	GC%	Tm
ad	cgatggcgtgtgccgtagcg	20	0.70	74.5
bc	cccgagcgttggacagccct	20	0.70	73.3
eg	caacggtggcgctcctgggg	20	0.75	76.5
fh	ctctcaggcggagggcgtcg	20	0.75	74.2

For matrix = Even:

Generate complementary edge sequences for all elements of value 1 from corresponding row and column sequences. From the 2nd matrix, elements of value 1 exist for intersections of *row c – column e* and *row d – column e*. From the 4th matrix, elements of value 1 exist for intersections of *row g – column j* and *row h – column j*. Therefore, the generated edges for the even matrices are as shown in Table 4:

Table 4 DNA Sequence for Edges (Matrix: Even)

Edges	Sequence	SeqLen	GC%	Tm
ce	cctgtcgggagttgccaccg	20	0.70	72.8
de	acggcatcgcgttgccaccg	20	0.70	77.8
gj	cgaggacccccccgcctta	20	0.75	75.2
hj	ctcccgcagcccccgcctta	20	0.75	76.4

Figure 3 shows the constructed edges for the Truncated Matrix.

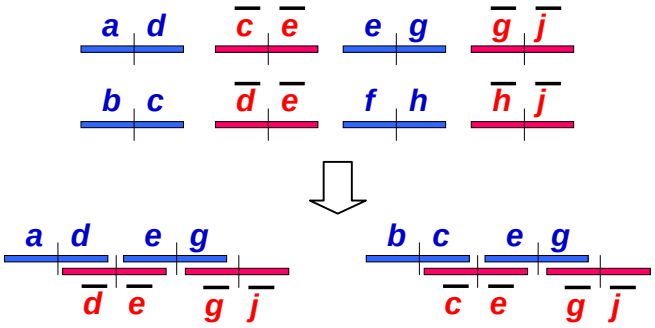

Figure 3 Constructed Edges for Truncated Matrix

3. EXPERIMENTAL RESULTS

Generated single stranded DNA sequences representing the constructed edges are poured into a single test tube T0 for generation of initial pool containing all possible solutions. Parallel Overlap Assembly method is utilized to allow the individual DNA strands to assemble and hybridize to their complementary strands. The strands will be extended with each denaturing, annealing and elongation cycles. At the end of the cycles, a formed path for solution of the problem will be constructed which can be extracted as output of the computation. The formed path consists of sequences containing initial vertex and terminal vertex sequences. We then allocate test tubes T1, T2, T3 and T4 to represent the intersections of *row a – column i*, *row a – column j*, *row b – column i* and *row b – column j* in the product matrix respectively. Contents of test tube T0 after the POA process are distributed into each test tube T1 – T4. Gel electrophoresis process is utilized to visualize the output of the computation. The gel electrophoresis is a technique which separates the DNA solution based on their size or sequence lengths. Therefore, the lengths of the formed paths in the test tubes T1 – T4 can be determined. From the captured image of gel electrophoresis process in Figure 4, the most left lane is the marker image for sequence lengths which are in base pair (b.p) unit for the constructed double stranded sequences of formed paths.

From the gel electrophoresis, test tubes T2 and T4 yield 50 b.p bands which are consistent with constructed paths of elements with value 1 in the product matrix representing the formed path from vertices *a – j* and *b – j*. In test tubes T1 and T3, highlighted bands show 40 b.p and 30 b.p strands indicating incomplete paths.

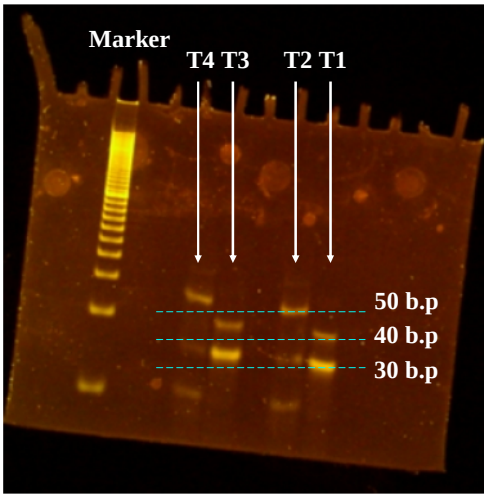

Figure 4 Results from gel electrophoresis process

3. CONCLUSIONS

In this work, we propose Truncated Matrices to reduce the material consumption in DNA computing. Using the original algorithm, assuming all 18 DNA sequence strands in Table 1 as set as 20 mer length sequences, the output of the computation will be of length 100 b.p. which is directly proportional to its sequence length from the initial to terminal vertex. However, using 20 mer length strands in Truncated Matrices will only generate a 50 b.p length output of formed path solution. The number of DNA strands for the computation is also reduced to 8 as shown in Table 3 and 4. The major constraints to successful implementation of the experiments come from in the instability of the generated DNA sequence strands. While simulated DNA sequences theoretically in compliance to expected temperatures and other parameters, the problem comes when actual DNA sequence strands differ in their compositions.

ACKNOWLEDGEMENTS

The authors would like to thank Universiti Malaysia Sarawak for the support.

REFERENCES

[1] Adleman L, (1994), "Molecular computation of solutions to combinatorial problems", Science, vol. 266, no. 5187, pp. 1021-1024.
[2] Amos M, (2004), Theoretical and Experimental DNA Computation, Natural Computing Series, Springer.
[3] Cox J C, Cohen D S & Ellington A D, (1999), "The complexities of DNA computation", Trends Biotechnology, 171, pp 151-154.
[4] LaBean, Reif M C & Seeman T H, (2003), "Logical computation using algorithmic self-assembly of DNA triple-crossover molecules", Nature, vol. 407, pp 493-496.
[5] Fu B & Beigel R, (1997), "A Comparison of Resource Bounded Molecular Computation Models, in Proceedings of the 5th Israel Symposium on Theory of Computing and Systems, pp 6-11.
[6] Bunow, (1995), "On the potential of molecular computing," Science, vol. 268, pp. 482-483.
[7] Rajaee N & Ono O, (2007), "Boolean Matrix Implementation with DNA computing", Proceedings of International Conference on Robotics, Vision, Information and Signal Processing (ROVISP), pp 36-39

[8] Rajaee N & Ono O, (2010), " Experimental Implementation of Boolean Matrix Multiplication with DNA Computing using Parallel Overlap Assembly", International Journal of Innovative Computing, Information and Control (IJICIC) ,vol. 6, no.2, pp. 591 - 599

[9] Rajaee N, Aoyagi H & Ono O, (2010), "Single Stranded DNA and Primers for Boolean Matrix Multiplication with DNA Computing", ICIC-EL International Journal of Research and Surveys ISNN1811-803X, pp 1067-1072

[10] Murphy R C, Deaton R, Stevens S E & Franceschetti R, (1997), "A new algorithm for DNA based computation", Proceedings of the IEEE International Conference on Evolutionary Computation, pp 207-212

[11] Kaplan P D, Ouyang Q, Thaler D S & Libchaber A, (1997), , "Parallel Overlap Assembly for the construction of computational DNA libraries", Journal of Theoretical Biology, vol. 188, pp 333-341

[12] Boneh D, Dunworth C & Sgall J, (1996), "On the computational power of DNA", Discrete Applied Mathematics, vol. 71, pp 79-94.

[13] Ogihara M & Ray A, (1996), "Simulating Boolean circuits on a DNA computer", Technical Report TR 631, University of Rochester.

Authors

Nordiana Rajaee received her BEng (Hons) from Electronic and Information Engineering from Kyushu Institute of Technology, Fukuoka, Japan in 1999, her MSc in Microelectronics in University of Newcastle Upon Tyne, United Kingdom in 2003 and PhD in DNA Computing from Meiji University, Japan in 2011.

Awang Ahmad Sallehin Awang Husaini received his BSc (Hons.) in Biochemistry and Microbiology from Universiti Putra Malaysia (UPM), Selangor, Malaysia in 1996 and PhD in Molecular Biology (Enzymology) from University of Manchester, Manchester, United Ki ngdom in 2001.

Sharifah Masniah Wan Masra received her BEng (Hons) in Telecommunication from Universiti Malaya, Kuala Lumpur, Malaysia in 2000 and MSc in Electronic from Cardiff University, Wales, UK in 2003