DNA Self-assembly Model for Matrix Addition Problem

Zhixiang Yin, Bosheng Song*
Department of Mathematics and Physics. Anhui University of Science & Technology, Huainan, China
Email: boshengsong@163.com

***Abstract*—The technology of DNA self-assembly has played an important role in the field of DNA computing and nanotechnology. Many small-scaled NP complete problems can be solved by self-assembly model. In this paper, we based on the addition of two numbers, and proposed the addition of two matrices of DNA Tile self-assembly model. The algorithm can be applied to add two elements in the corresponding positions automatically, and then in top line of the assembly appears the final results. Theoretical analysis shows that the model can solve the matrix addition operation of any order.**

***Index Terms*—DNA computing; NP complete problem; matrix addition; self-assembly**

I. INTRODUCTION

DNA computing is a new kind of information processing pattern, which is based on biochemical reaction with DNA molecules, bio-enzyme and so on being the most basic materials. DNA computing was first put forward by Adlema[1,2] in 1994. He solved the problem of a seven vertices Hamilton road with DNA molecule, and successfully conducted the biochemical experiments. The significance of the experiment was that the application of a brand-new medium for calculation provides a new computing method to solve difficult problems, which can not be solved by traditional electronic computers in multinomial time. Thus he created a totally new field of computing, which attracted many scholars' interest. If we think of a strand of DNA as a sequence of characters chosen from a four-letter alphabet, then it is clear that we may use it to represent information in the same way that electronic computers use a two-letter (binary) alphabet. The general principle of extant DNA computation is to first generate a random set of strands in which solutions to a given problem occur with high probability. The computation then proceeds by using standard manipulation techniques to isolate those that are solutions. In 1995, Lipton[3] established a computing model rested on the SAT problems of DNA computing. In 1996, Roweis[4]and some other scholars put forward a new kind of DNA computing model--Sticking model by further research of the mechanism of biological calculation. In the same year, Frank[5] advanced an addition model of DNA computing and published in the science. Professor Sakamoto[6],from Tokyo University, created a hairpin form of DNA molecular for DNA computing model. In 2002, Braich[7] and others successfully solved a SAT problem with 20-variables.This was the largest scale of NP complete problems which were solved by using the DNA computing model at that time. Theoretical analysis showed that, the highly concurrence and huge storage capacity of DNA computing made it an effective method to solve the NP complete problems.

DNA tile self-assembly is looked forward to many applications in different fields. More and more scientists have paid their attention on this field. Therefore, DNA computation based on self-assembly has been playing a significant role in bio-molecular computing. Mao [8] came up with triplecrossover to execute four steps of logical (cumulative XOR) operations on a string of binary bits. These kinds of tiles are more stable and rigid which ensure that the process of tile assembly can achieve our desired computation. Barish et al.[9] have proposed an algorithmic self-assembly to perform two primitive computations: copying and counting, and experimentally demonstrated the potential of algorithmic self-assembly to create complex nano-scale patterns. Fujibayashi et al. [10] have used DNA tiles and DNA origami to grow crystals containing a cellular automaton pattern and proved that programmable molecular self-assembly may be sufficient to create a wide range of complex objects in one-pot reactions. Dwyer and his research group [11] have proposed two architectures that are enabled by self-assembly for implementation of DNA computers solving highly demanding computational problems.

In addition, a variety of researches have emerged in this field and applied this biological technique to implementing more intricate computation. NP-complete problems have been considered as a kind of problems which are hard to solve by the conventional electronic computer. In 2000, Lagoudakis et al.[12] presented an algorithmic design for solving the SAT problem using 2-dimensional DNA self-assembly. His design and encoding of the algorithm in a general way separated the algorithm from the data and minimized the dependence on particular instance. Jonoska and McColm [13] have developed an algebraic representation of the self-assembly process and use it to prove that his model of self-assembly precisely captures NP-computability under certain conditions.

In recent 10 years, DNA self-assembly has been widely developped in the fields of molecular calculation, biophysics, nanometer technical, etc. In 1989, Seeman first proposed a crisscross structure as the basic element of DNA self-assembly. In 1998, Winfree[14-16], based on Seeman's molecular nanometer structure, had the idea

doi:10.4304/jcp.6.4.698-704

that using the procedure of the Tile model to construct the nanometer structure can help to achieve calculation. And they first constructed the simple of two dimensional tiles using DNA strands, including doublecrossover molecules and triple-crossover molecules, to demonstrate the feasibility of computing through the self-assembling of DNA tiles. In 2003, Hao[17] designed the crystal of DNA self-assembly, which included DNA barcode patterns. It was the first time to construct the non-periodic pattern by making use of DNA self-assembly. In 2004, Rothemund[18,19] took advantage of DX to design Sierpinski triangle self-assembly, and used the automata theory of one-dimensional cellular to realize XOR operations. In 2007, Brun[20-23], the student of Adleman, created the addition and multiplication model with the application of Tile self-assembly. In 2008, Qian[24] used DNA self-assembly to realize the logic circuit with "and" and "not" gates, which could be employed to design molecular circuit. In 2009, Zhang[25] achieved the subtraction and division with the application of DNA self-assembly. In the same year, Cheng[26] proposed a method on the base of DNA Tile self-assembly algorithm to solve the problem of subset accumulations.

II. FORMALIZATION OF TILE SELF-ASSEMBLY

Self-assembly is a process that is ubiquitous in nature. Systems form on all scales via self-assembly: atoms self-assemble to form molecules, molecules to form complexes, and stars and planets to form galaxies. One manifestation of self-assembly is crystal growth: molecules self-assembling to form crystals. Crystal growth is an interesting area of research for computer scientists because it has been shown that, in theory, under careful control, crystals can compute.

Building on classical Wang tiling models[27] dating back to the 1960s, Rothemund and Winfree in 2000 proposed an elegant discrete mathematical model for complexity theoretic studies of self-assembly known as the Tile Assembly Model. In this model, DNA tiles are treated as oriented unit squares (tiles). Each of the four sides of a tile has a glue with a positive integral strength. Assembly occurs by accretion of tiles iteratively to an existing assembly, starting with a distinguished seed tile. A tile can be "glued" to a position in an existing assembly if the tile can fit in the position such that each pair of abutting sides of the tile and the assembly have the same glue and the total strength of the glues is greater than or equal to the temperature, a system parameter. Research in this field largely focuses on studying the complexity of and algorithms for (uniquely and terminally) producing assemblies with given properties, such as shape. It has been shown that the construction of $n \times n$ squares has a program size complexity (the minimum number of distinct types of tiles required) of $\Theta(\frac{\log n}{\log \log n})$ [28]. The upper bound is obtained by simulating a binary counter and the lower bound by analyzing the Kolmogorov complexity of the tiling system. This model was later extended by Adleman et al. to include the time complexity of generating specified assemblies. Later work studies various topics, including combinatorial optimization, fault tolerance, complexity problems and topology changes, in the standard Tile Assembly Model as well as some of its variants [29-40].

DNA Tile self-assembly model has been fully defined in Ref [20] and the definitions here are similar to those. It is necessary to restate them here to assist the readers to understand the following discussions. It is defined that the glue is used to represent binding domain of each tile. Each single-stranded overhang of a tile is designed different so that the glue corresponding to each side is different. Formally, each tile will be covered with a "specific" glue on its east, south, west and north side. It may stick to another tile when the binding domain on the abutting sides of those tiles match when total strength of all the abutting domains on the tile equals or exceeds the current temperature. By specific, it is defined that each type of glue will have a set of glues to which it can stick and which can stick to it and disjoint set of glues to which it cannot stick and which cannot stick to it.

Tile self-assembly model mainly consists of four parts :

- basic Tile types: the basic Tile types used to construct various calculation operator, stored the value of operation,to finish all kinds of calculation.
- A strength function: used to define two adjacent Tile at any two domains intensity.
- Seed Tile: used to define the beginning and end for a self-assembly.
- Parameter τ : used to express the thermal dynamic parameters, only when the domain strength sum between every two Tile greater than a given parameter τ , the whole assembly steady

Formally, let Σ be a finite alphabet of binding domains such that null $\in \Sigma$. A tile over a set of binding domains Σ is a 4-tuple$< \sigma_N, \sigma_E, \sigma_S, \sigma_W > \in \Sigma^4$. A position is an element of $\mathbb{Z}^2$. The set of directions $D = \{N, E, S, W\}$ is a set of four functions from positions to positions, i.e. $\mathbb{Z}^2$ to $\mathbb{Z}^2$ such that for all positions (x, y), $N(x, y) = (x, y+1)$, $E(x, y) = (x+1, y)$, $S(x, y) = (x, y-1)$, $W(x, y) = (x-1, y)$. If (x', y') and (x, y) are next to each other, and only if the positions of (x', y') and (x, y) meet the conditions of $\exists d \in D, d(x, y) = (x', y')$. For a tile t, for $d \in D$, $bd_d(t)$ as the binding domain of tile t on d^{l} s side. A special tile empty = <null, null, null, null> represents the absence of all other tiles.

A strength function g : $\Sigma \times \Sigma \to R$, $\forall \sigma \in \Sigma$, $g(null, \sigma) = 0$. It is common to assume that $g(\sigma, \sigma') = 0 \Leftrightarrow \sigma \neq \sigma'$. This simplification of the model implies that the abutting binding domains of two tiles have to match to bind.

Let T be a set of tiles containing the empty tile. A configuration of T is a function A : $\mathbb{Z} \times$

$Z \to T$. $A(x,y) \neq empty$, $(x,y) \in A$. There is only a finite number of distinct positions $(x,y) \in A$.

Finally, a tile system $\mathbb{S}$ a 4-tuple $\langle S,T,g,\tau \rangle$, where S is a seed-configuration; T is a finite set of tiles containing empty; g is a strength function; $\tau \in N$ is the temperature.

If A is a configuration, then within system $\mathbb{S}$, a tile t can attach to A at position (x,y) and produce a new configuration A' iff:

(1) $(x,y) \notin A$ *and*

(2) $\sum_{d \in D} g(bd_d(t), bd_{d^{-1}}(A(d(x,y)))) \geq \tau$, *and*

(3) $\forall (u,v) \in Z^2$, $(u,v) \neq (x,y) \Rightarrow A'(u,v) = A(u,v)$, *and*

(4) $A'(x,y) = t$.

That is, a tile can attach to a configuration only in empty positions and only if the total strength of the appropriate binding domains on the tiles in neighboring positions meets or exceeds the temperature τ.

Given a tile system $\mathbb{S}=\langle S,T,g,\tau \rangle$, a set of tiles T, and a seed configuration $S : \mathbb{Z}^2 \to S$, if the above conditions are satisfied, one may attach tiles of T to S. Configurations produced by repeated attachments of tiles from T are said to be produced by $\mathbb{S}$ on S. If this process terminates, then the configuration achieved when no more attachments are possible is called the final configuration. If for all sequences of tile attachments, all possible final configurations are identical, then $\mathbb{S}$ is said to produce a unique final configuration on S.

III. Two Numbers Add System

A. Preliminary

This paper establishes a tile assembly model on the basis of triple-crossover tile molecules. As shown by Brun [20], a tile can attach to a configuration only in empty positions and only if the total strength of the appropriate binding domains on the tiles in adjacent positions meets or exceeds the temperature t. Herein, we give some assumptions that $g=1$ and $t=2$. Consequently, it means that a tile t can attach only at positions with matching binding domains on the tiles in at least two adjacent positions.

A Lot of research has demonstrated that tile assembly is a powerful technique for performing DNA-based computation. Here, we devote to theoretically developing a tile assembly system model which is used to accomplish the matrix addition problem. We mainly consider the triple-crossover molecule as the basic computational unit; their molecular structure is illustrated in Fig.1. The molecule contains four strands (shown in red, green, blue, purple) that self-assemble through Watson-Crick base pairing to produce three double helixes in a roughly planar plane. Each double helix is connected to the adjacent double helical domains at two points where their strands cross over between them. The ends of the central double helix are closed by hairpin loops. Our design is described at the algorithmic level so that we abstract each DNA tile as a square for simplicity with labels at each side(see Fig.1). Each alphabetic label formally indicates a parallel kind of sticky end. Two sticky ends that can be attachable in the Watson-Crick sense and ligation are represented by identical labels. Each tile can have from one to four labels.

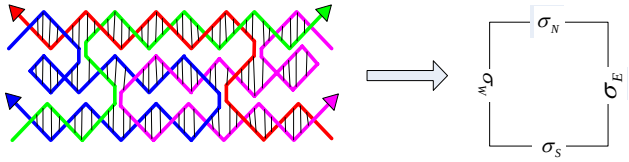

Fig.1.Triple-crossover tile and its abstracted representation. The structure in the left contains four strands (shown in red, green, blue and purple) that self-assemble through Watson-Crick base pairing to produce three double helices in a roughly planar arrangement. The ends of the central double helix are closed by hairpin loops, but the other helices can terminate in sticky ends containing information that directs the assembly of the tiles. For simplicity, the right square is used to abstract the structure of the left tile. The four sticky ends are shown by σ_i where $i \in \{E,S,W,N\}$, and only the sticky ends of the adjacent tiles which owned the same information that can attach to each other.

Computation by self-assembly is the spontaneous selfordering of substructures into superstructures driven by annealing of Watson-Crick base-pairing DNA sequences. Computation by DNA tile self-assembly entails the building up of superstructures from starting units such that the assembly process itself performs the actual computation. DNA tile self-assembly is also a highly parallel process, where many copies of different molecules bind simultaneously to form intermediate complexes. One might be seeking to construct many copies of the same complexes at the same time, as in the assembly of periodic 1D or 2D arrays; Alternatively, one might wish to assemble in parallel different molecules, as in DNA-based computation, where different assemblies are sought to test out the combinatorics of the problem.

A sequential or deterministic process of DNA tile self-assembly has three highly parallel instruction steps [40]. The first one is molecular recognition: elementary molecules selectively bind to others. The second is growth: elementary molecules or intermediate assemblies are the building blocks that bind to each other following a sequential or hierarchical assembly. The cooperativity and non-linear behavior often characterize this process. The third way is termination: a built-in halting feature is required to specify the completion of the assembly. In practice, their growth is interrupted by physical and/or environmental constraints. DNA tile self-assembly is a time-dependent process and because of this, temporal information and kinetic control may play a role in the process before thermodynamic stability is reached. The process of DNA tile self-assembly has three characters [41]: it is a timedependent process and because of this, temporal information and kinetic control may play a role in the process before thermodynamic stability is reached. And the molecular self-assembly is also a highly parallel process, where many copies of different molecules bind

simultaneously to form intermediate complexes. Another characteristic of a molecular self-assembly is that the hierarchical build-up of complex assemblies allows one to intervene at each step, either to suppress the following one, or to orient the system towards a different pathway.

B. Two Numbers Add System

Here we use the L-configure to calculate the addition operator, the method use a size $\Theta(1)$ tile set. It is important to construction the matrix addition system, the following will give a detailed presentation.

Encodes two numbers in binary: $a = 83 = 1010011_2$, $b = 35 = 100011_2$. Then we use of DNA tile self-assembly to solution $a+b$. The below show are the abstract tile types and the basic tile type, and an example of the seed configuration and finail configuration(Fig.1).

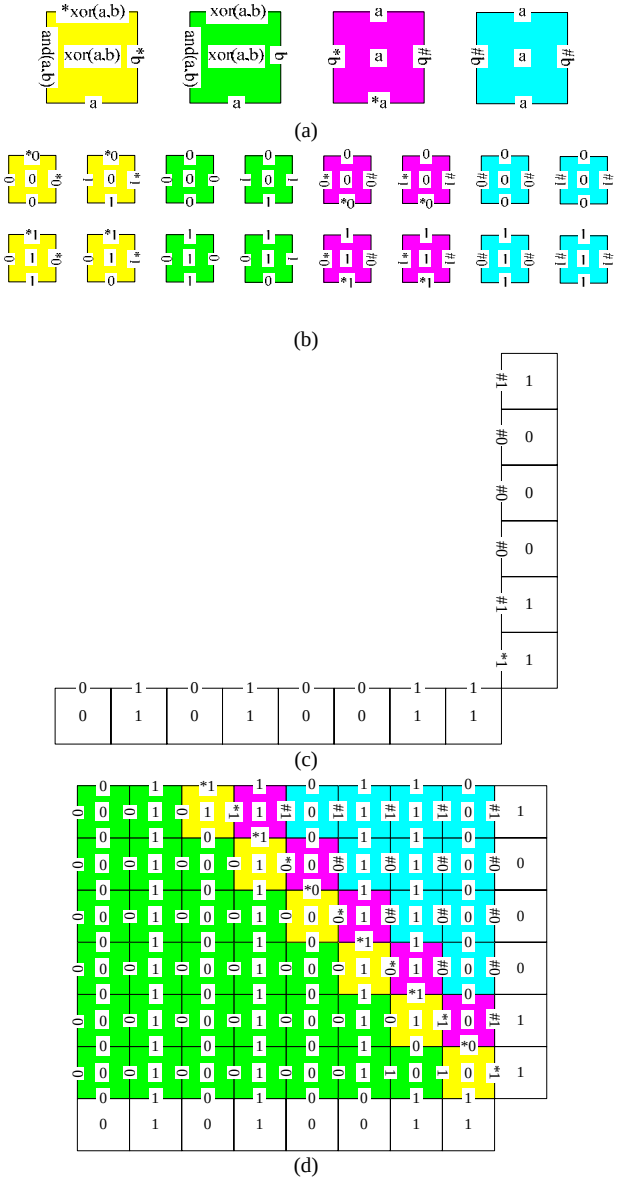

Fig 2. An example for addition system The system computes $83+35$. The tiles have two input sides (east and south)and two output sides (west and north). (d) shows the final stage of the addition system.

Theorem 1: Let $\Sigma_{16} = \{0,1,\#0,\#1,*0,*1\}$, $g_{16} = 1$, $\tau_{16} = 2$, T_{16} is a tile set for Σ_{16}, show as Fig1(b). Then $\mathbb{S}_{+16}$= $< S_{16}, T_{16}, g_{16}, \tau_{16} >$ computes the function $f(a,b) = a+b$.

In L-configure, there are four abstract tile types: yellow, green, magenta and blue four colors. Blue tiles are used to convey information; magenta tiles that perform two jobs: propagating the information up just as the blue tiles and guiding the yellow diagonal line. Green and yellow tiles to perform addition operation. As for addition system, one of the input numbers is coded on the bottom row and the second input number on the rightmost column of an L-configuration. The adder adds one bit of the column number to the row number, per row. The i th bit has to be added at the ith position, and the system uses the yellow diagonal line to compute that position. Each tile has two input sides (south and east) and two output sides (north and west). The north side is the value of the tile and the west side is the carry bit. The 1 or 0 in the middle of the tile t is that tile's $v(t)$ value.

Fig.2(c) shows a sample seed configuration which encodes two numbers in binary: $1010011_2 = 83$, $100011_2 = 35$. Just as before, Number 83 is encoded on the bottom row and number 35 is encoded on the rightmost column. There are four tiles in Σ_{16}, the 1 and 0 tiles for each of the inputs. Note that at the start, only one tile may attach to this configuration because $\tau = 2$ and there is only a single corner. Therefore, $\mathbb{S}_{+16}$ produces a unique final configuration. Fig.2(d) shows the final configuration for the example of $83+35$, with the solution encoded on the top row. The row reads $1110110_2 = 118 = 83+35$. Just as before, because the sum of two n-bit numbers may be as large as $n+1$ bits, the row input needs to be padded with a single extra 0 tile as its most significant bit. The column input does not need this padding.

IV. MATRIX ADDITION SYSTEM

Through the above description, we are clear that the addition system of tile self-assembly. But for the sum of two matrix: A_{mn} and B_{mn}. There are $m \times n$ pairs numbers for addition. In order to read out these results, we need to add some separators. Each of the matrix in the corresponding location are separated with separator. So it can read the result easily.

The following we will given a simply example to illuminate the matrix addition system of DNA tile self-assembly. The calculation tiles are the same as above(Fig.2(b)). The tiles for separate as show Fig3.(a). Note here, two matrices used in the separators can not be the same. Otherwise it would causeambiguity in calculation. So it can't get the right result. We take a 2×2 matrix addition as an example to show the calculation process. The matrix addition is as follow.

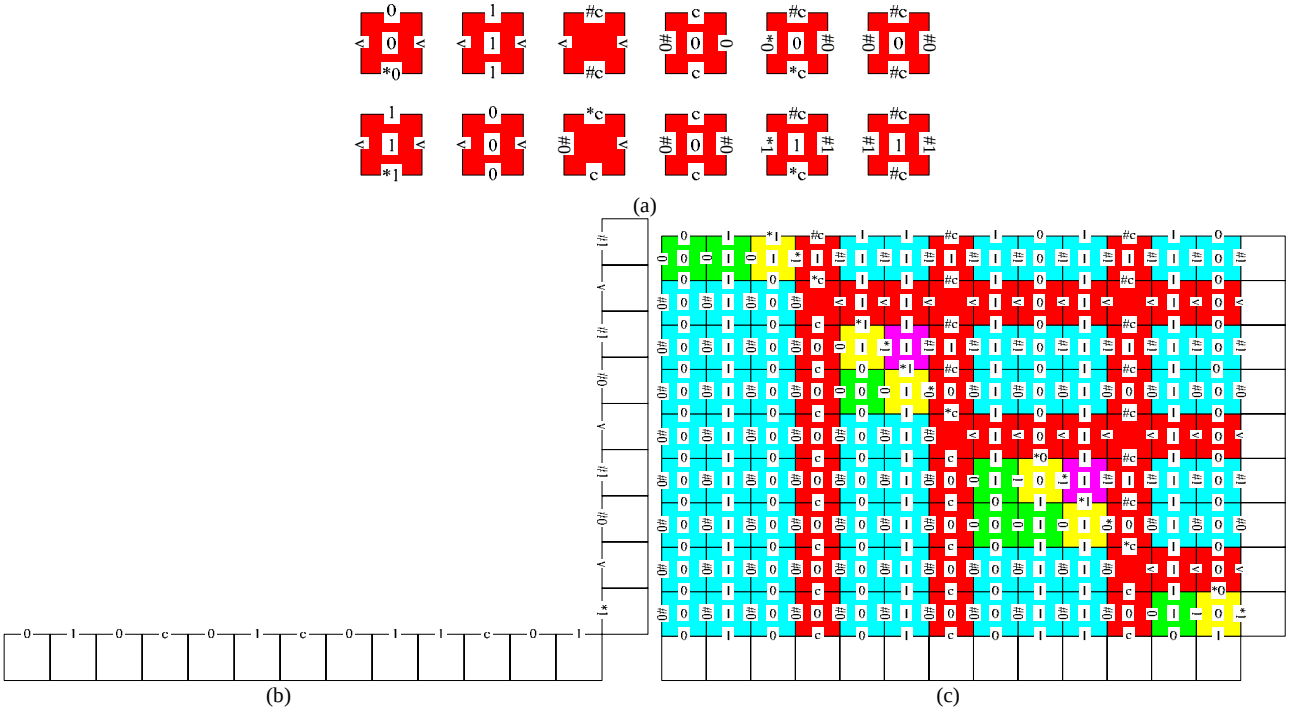

Fig.3 An example for matrix addition system. (a) The basic tile types of the matrix addition system. (b) The seed configuration for matrix addition system. (c) The final stage for matrix addition system.

$$\begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix}$$

Fig.3(a) 12 kinds of tiles, they perform two jobs: propagating the information up and chansmission information. For example, after a_{11} add b_{11}, through the red tiles to transmit information up. Fig.3(b) give a seed configuration for two matrix addition. Just as before, in front of each number(the bottom row), needs to be padded with a single extra 0 tile as its most significant bit. The column input does not need this padding. Fig.3(c) shows the final configuration. From Fig.3(c) we can see clearly that the four partitioned matrices along the main diagonal rectangular, they are for add operation. It involves yellow, magenta and green tiles. Both the red and blue tiles are for information transmission. The final result showed at the top line. From the right to left are the values of matrix at each position. The value are: $2=10_2$, $5=101_2$, $3=11_2$, $3=11_2$.

In theory, the model can be calculation to any order of the addition operation of two matrices. Just as increases the number of the matrix and the order, the assembly will be increased also. In the paper, the elements of value are relatively small. It should also be noted that the matrix addition model is only applicable to each element of the matrix are non-negative, for the more general case of matrix addition model also need to study in the future.

V. Complexity Analysis

As for matrix addition model, the self-assembly time is relevant with the size and the number of element contain in the matrix. Let the largest element in the matrix is n -binary, and the matrix contain $n \times n$ elements. Therefore, the time complexity of the assembly is:

$$T = n \times n \times n = \Theta(n^3).$$

The space complexity of the system S is the area of the assembly:

$$S = [(n+1)n^2 + n^2 - 1][n^3 + n^2 - 1] = \Theta(n^6).$$

Finally, the type of the required tiles are:
Border tiles: 8;
Calculated tiles: 16;
Tiles for separator and information transfer: 12;
Total tiles: $8+16+12=36=\Theta(1)$.

VI. Conclusion

The paper is based on the addition model, and we proposed the matrix addition model by DNA tile self-assembly. The elements of the matrix are encoding by binary, the encoding sequence of elements in the matrix are from each row, and corresponding to the seed configuaration are from right to left, bottom to top. The results display in the top row of the system. From right to left is the results of each elements in the matrix. In theory, this model can calculate any order of the matrix addition operation.

As DNA tile self-assembly suffers from high error rates, the possible sources of errors are, either an error in constructing the tiles, or an erroneous binding of tiles. Methods of error control and error correction may be used to decrease the error rates in the computation of DNA tile self-assembly model. Many experimental results in DNA tile self-assembly have not appealed to the advantages of crystal growth; however, these early works on the fundamentals of self-assembly and the

physical experimental evidence of actual DNA tile crystals suggest a bright future for DNA tile self-assembly. The field of nanotechnology holds tremendous promise, but many technical hurdles will have to be overcome before algorithmic DNA tile self-assembly can be developed into a practical commercial technology. If the molecules and supramolecules can be controlled at will, then it may be possible to achieve vastly better performance for computers and memories. So we can see that the DNA tile self-assembly model has various applications in many fields and it also might open up a host of other applications in materials science, medicine, biology and other ways.

ACKNOWLEDGMENT

We thank every authors appeared in the references. This Project supported by CNSF (Grant number: 30570431, 60873144); Science Foundation of Educational government of Anhui Province of China(KJ2007B173); Program for Excellent Talents in Anhui , Program for New Century Excellent Talents in University(NCET-06-0555); National High Technology Research and Development Program of China (2006AA01Z104);Item of Special Science Research Plan from Shaanxi Province Education Office(08JK313)and Open Foundation of ISN.

REFERENCES

[1] L. Adleman, "Molecular Computation of Solution to Combinatorial Ploblems," Science, vol. 66, 1994, pp. 1021-1024.
[2] L. Adleman, "Towards a mathematical theory of self-assembly," Technical Report 00-722. Department of Computer Science, University of Souther California, Los Angeles, CA, 2000.
[3] R J. Lipton, "DNA Solution of Hard Computation Problem," Science, vol. 268, 1995, pp. 542-545.
[4] E. Roweis, R. Winfree, and Burgoyne, et al, "A stick Based Model for DNA Computation," DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 44, 1999, pp. 1-27.
[5] G. Frank, F. Makiko, and B. Carter, "Making DNA add," Science, vol.273, 1996, pp. 220-223.
[6] K. Sakamoto, H. Gouzu, and K. Komiya, et al, "Molecular computation by DNA hairpin formation," Science, vol. 288, 2000, pp. 1223-1226.
[7] R S Braich, N. Chelyapov, and C. Johnson, et al, "Solution of a 20-Variable 3-SAT Problem on a DNA Computer," Science, vol. 296 2002, pp. 430-434.
[8] C. D. Mao, T. LaBean, J. Reif and N. C. Seeman, "Logical computation using algorithmic self-assembly of DNA triple-crossover molecules," Nature, vol. 407, 2000, pp. 493-496.
[9] R. D. Barish, P. W. K. Rothemund and E. Winfree, "Two computational primitives for algorithmic self-assembly: copying and counting," Nano Lett., vol. 5, 2005, pp. 2586-2592,.
[10] K. Fujibayashi, R. Hariadi, S. H. Park, E. Winfree and S. Murata, "Toward reliable algorithmic self-assembly DNA tiles: a fixed-width cellular automaton pattern," Nano Lett., vol. 8, 2007, pp. 1791-1797.
[11] C. Dwyer, J. Poulton, R. Taylor and L. Vicci, "DNA self-assembled parallel computer architectures," Nanotechnology, vol. 15, 2004, pp. 1688-1694.
[12] M. G. Lagoudakis and T. H. LaBean, "2D DNA self-Assembly for Satisfiability," DIMACS Series in DISCRETE Mathematics and Theoretical Computer Science, 2000, pp. 139-152.
[13] N. Jonoska and G. L. McColm, "A computational model for selfassembling flexible tiles," Berlin Heidelberg, Germany: Springer Verlag, 2005, pp. 142-156.
[14] E. Winfree, "Algorrithmic Self-Assembly of DNA," Ph.D Thesis, California Institute of Technology, 1998.
[15] E. Winfree. "Design and self-assembly of Two-Dimensional DNA Crystals," Nature, vol. 394, 1998, pp. 1223-1226.
[16] E. Winfree, "Simulations of computing by self-assembly of DNA," Technical Report CS-TR:1998:22, California Insitute of Technology, Pasadena, CA, USA, 1998.
[17] H. Yan, T H Labean, and L. Feng, et al, "Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices," Proceedings of the National Academy of Science, vol. 100, 2003, pp. 8103-8108.
[18] P W K Rothemund, N. Papadakis, and E. Winfree, "Algorithmic self-assembly of DNA Sierpinski triangles," PloS Biology, vol. 2, 2004, pp. 2041-2053.
[19] P W K Rothemund, E, Winfree, "The program-size complexity of self-assembled squares," in: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing , ACM Press, 2000, pp. 459 – 468.
[20] Y. Brun, "Arithmetic computation in the tile assembly model: Addition and multiplication," Theoretical Computer Science, vol. 378, 2007, pp. 17-31.
[21] Y. Brun, "Nondeterministic Polynomial Time Factoring in the tile Assembly Model," Theoretical Computer Science, vol. 395, 2008, pp. 3-23.
[22] Y. Brun, "Path finding in the tile assembly model," Theoretical Computer Science, vol. 410, 2009, pp. 1461-1472.
[23] Y. Brun, "Solving NP-complete problems in the tile assembly model," Theoretical Computer Science, vol. 395 2008, pp. 31-46.
[24] L. Qian, E. Winfree, "A simple DNA gate motif for synthesizing large-scale circuits," In: DNA 14, Prague, 2008, pp. 1-13.
[25] X. Zhang, Y. Wang, and Z. Chen, et al, "Arithmetic computation using self-assembly of DNA Tiles: subtraction and division," Progress in Natural Science, vol. 19, 2009, pp. 377-388.
[26] Z. Cheng, J. Xu, and Y. Huang, et al, "Algorithm of solving the Subset-Product Problem Based on DNA Tile Self-assembly," Journal of Computational and Theoretical Nanoscience, vol. 6, 2009, pp. 1161-116.
[27] H. Wang, "Proving theorems by pattern recognition Ⅱ," Bell Systems Technical Journal, vol. 40, 1961, pp. 1-42.
[28] L. Adleman, Q. Cheng, A. Goel, and M.D. Huang, "Running time and program size for self-assembled squares," in: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, ACM Press, 2001, pp. 740-748.
[29] L. Adleman, Q. Cheng, A. Goel, M.D. Huang, D. Kempe, P.M. de Espans, and P.W.K. Rothemund, "Combinatorial optimization problems in self-assembly," in: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, ACM Press, 2002, pp. 23 – 32.
[30] G. Aggarwal, M.H. Goldwasser, M.Y. Kao, and R.T. Schweller, "Complexities for generalized models of self-assembly," in: Proceedings of 15th Annual ACMSIAM Symposium on Discrete Algorithms, SODA, ACM Press, 2004, pp. 880 – 889.

[31] H.L. Chen, Q. Cheng, A. Goel, M.D. Huang, and P.M. de Espanes, "Invadable self-assembly: Combining robustness with efficiency," in: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2004, pp. 890 – 899.
[32] H.L. Chen, A. Goel, "Error free self-assembly using error prone tiles," in: DNA Based Computers 10, 2004, pp. 274 – 283.
[33] Q. Cheng, A. Goel, and P. Moisset, "Optimal self-assembly of counters at temperature two," in: Proceedings of the First Conference on Foundations of Nanoscience: Self-assembled Architectures and Devices, 2004.
[34] M. Cook, P.W.K. Rothemund, and E. Winfree, "Self-assembled circuit patterns," in: DNA Based Computers 9, in: LNCS, vol. 2943, 2004, pp. 91 – 107.
[35] K. Fujibayashi, S. Murata, "A method for error suppression for self-assembling DNA tiles," in: DNA Based Computing 10, 2004, pp. 284 – 293.
[36] M. Kao, R. Schweller, "Reduce complexity for tile self-assembly through temperature programming," in: Proceedings of 17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, ACM Press, 2006.
[37] J.H. Reif, S. Sahu, and P. Yin, "Compact error-resilient computational DNA tiling assemblies," in: Proc. 10th International Meeting on DNA Computing, 2004, pp. 248 – 260.
[38] S. Sahu, P. Yin, and J.H. Reif, "A self assembly model of time-dependent glue strength," in: Proc. 11th International Meeting on DNA Computing, 2005, pp. 113 – 124.
[39] R. Schulman, E. Winfree, "Programmable control of nucleation for algorithmic self-assembly," in: DNA Based Computers 10, in: LNCS, 2005.
[40] D. Soloveichik, E. Winfree, "Complexity of self-assembled shapes," in: DNA Based Computers 10, in: LNCS, 2005.
[41] A. Carbone, N.C. Seeman, "Molecular Tiling and DNA Selfassembly," Springer-Verlag Berlin Heidelberg, LNCS 2950, 2004, pp. 61–83.

Zhi-Xiang YIN was born in China, in 1966. He received his M.S. degree in the school of science from Nanjing Normal University, Nanjing, China, in 1991 and his Ph.D. degree in the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2003, respectively.

He is a supervisor of M.S. candidates and also the dean of Department of Mathematics and Physics, Anhui University of Science & Technology, Huainan, China. He has published many articles in the domestic and foreign magazines such as "Journal of Chemical Information and Computer Science", "Biosystems", "Journal of Computational and Theoretical Nanoscience" and so on.

Prof Yin's current research interests include combinatorial and optimization, DNA computing and protein structure prediction.

Bo-Sheng SONG was born in China, in 1986. He received his B.S. degree in the Department of Mathematics and Physics, Anhui University of Science & Technology, Huainan, China, in 2008. He is currently working toward a M. S. candidate in the same Department of the University.

His current research interests include combinatorial and optimization, DNA computing.