Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/072L48

Fast parallel algorithm to the minimum edge cover problem based on DNA molecular computation

Zhaocai Wang[1], *Dongmei Huang*[1] *and Chengpei Tang*[2,*]

[1]College of Information, Shanghai Ocean University, Shanghai 201306, P. R China
[2]School of engineering, Sun Yat-sen University, Guangzhou 510006, P. R. China

Received: 28 Oct. 2012, Revised: 25 Feb. 2013, Accepted: 27 Feb. 2013
Published online: 1 Jun. 2013

Abstract: The minimum edge cover (MEC) problem is to find a smallest edge subset in a given undirected and simple graph, that every vertice in the graph at least belongs to one edge of the subset. It is a vitally important NP-complete problem in theory of computation and applied mathematics, having numerous real life applications. It can be difficultly solved by the electronic computer in exponential level time. In previous studies DNA molecular operations usually be used to solve NP-complete continuous path search problems (for example HPP, travelling salesman problem), rarely for NP-hard problems with discrete vertices or edges solutions result, such as the minimum edge cover problem, graph colouring problem and so on. In this paper, we present a DNA algorithm for solving the MEC problem with DNA molecular operations. For an undirected and simple graph with n vertices and m edges, we reasonably design fixed length DNA strands representing vertices and edges of the graph, take appropriate steps and get the solutions of the MEC problem in proper length range using $O(n^2)$ time. We theoretically proved the algorithm and simulate the DNA experiment to get correct solution of the ensample. We extend the application of DNA molecular operations and simultaneity simplify computational complexity of NP-complete problem.

Keywords: DNA computing; The minimum edge cover problem; Adleman-Lipton model; NP-complete problem

1 Introduction

NP problems are a class of mathematical problems which have most likely exponential complexity of computation, with no efficient algorithm having been found yet [1]. Meanwhile DNA computation has emerged in the last twenty years as an exciting new research field at the intersection of computer science, biology, engineering, and mathematics. Huge storage capacity and massive parallelism are two important advantages of DNA computation. DNA computing can execute billions of operations simultaneously. The massive parallelism of DNA computing comes from the large number of molecules which chemically interact in a small volume. DNA also provides a huge storage capacity since they encode information on the molecular scale. So DNA has a great application prospect for having wide range of abundant resources. The notion of performing computations at a molecular level was only realized in 1994, Adleman [2] presented an idea of solving the Hamiltonian path problem with n vertices in $O(n)$ steps using DNA molecules. Since then the field has blossomed rapidly, with significant theoretical and experimental results being reported regularly. Lipton [3] demonstrated that Adleman's experiment could be used to figure out the NP-complete satisfiability (SAT) problem (the first NP-complete problem). In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems [4-10].

However, most of the previous works in DNA computing are concentrated on solving the path search problems that the optimum results are continuous head-to-tail ligation edges or vertices sets. For example, Lee [11] first designs different length's strands representing paths values and cities, takes molecular operations to generate strands standing for all possible paths, then uses biochemical techniques, such as denaturation temperature gradient polymerase chain reaction and temperature gradient gel, to get the optimum solutions of the traveling salesman problem. To solve the shortest path problem, Narayanan [12] respectively

* Corresponding author e-mail: zcwang@yahoo.cn, tchengp@mail.sysu.edu.cn

carries out DNA reaction to get the strands for a list of series paths, then chooses the shortest length strands as the solution through DNA biotechnologies. The previous researches have some insufficient factors. One is that the strands for the possible paths are usually very long, Whereas too long DNA strands can lead to error-prone in annealing and separation procedures using modern biotechniques. The other is that previous research problems are all optimum path search problems, so that the possible solutions can be relatively easily represented by DNA strands. While the MEC problem is a discrete edge set problem with discontinuous path. So representation discrete data with DNA molecule is an important issue to expand the capability of DNA computing so as to solve many optimization problems.

The minimum edge cover problem is a problem of central importance in mathematical graph theory and computational sciences. It is intractable to solve. The earliest research on it traces back to 1950s. Motwani and Naor proved that MEC problem is NP-hard by showing the equivalence between the vertex coloring problem and the MEC problem. In 1972, Liberti et al. first presented an integer programming formulation for the MEC and also proposed heuristic algorithms for it. The authors used an integer programming solver to compute the optimal solutions for small instances to make comparison with their heuristics. Since then, No better algorithm has been derived up to now. However previous research work solve the MEC problem in exponential level time. With the increasing vertex number of n in graph, solving MEC problem will become more and more impractical by the previous algorithm. In this paper, a DNA procedure proposed by Adleman [2] and Lipton [3] is introduced for figuring out solutions of the minimum edge cover problem: Given an undirected and simple graph $G=(V,E)$ with a vertex set $V=\{v_1,v_2,\cdots,v_n\}$ and edge set $E=\{e_{i,j}|1\leq i<j\leq n\}$, a edge cover is a subset $E^{'}\subseteq E$ such that for any vertex $v_i\in V$ at least belongs to one edge $e_{i,j}$ of the subset $E^{'}$. A edge cover $E^{'}$ is to be a minimum edge cover of graph G, if for any edge subset $E^{''}\subseteq E$ with $|E^{'}|\leq|E^{''}|$. For instance, the undirected and simple graph G in Fig. 1 defines such a problem. It is not difficult to find that the edge subset $\{e_{1,2},e_{3,5},e_{4,6}\}$ is the solution to the minimum edge cover problem for graph G in Fig. 1.

The rest of this paper is organized as follows. In Section 2, the Adleman-Lipton model is introduced in detail. Section 3 uses a DNA molecular algorithm for solving the minimum edge cover problem. Section 4 proved DNA algorithm complexity and feasibility. We get conclusions in Section 5.

2 The Adleman-Liption Model

A DNA(deoxyribonucleic acid) is a polymer, which is strung together from monomers called

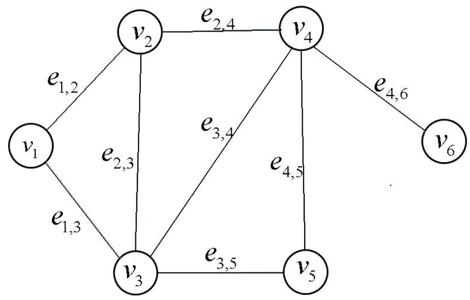

Fig. 1. An undirected and simple graph G with 6 vertices and 8 edges

deoxyribonucleotides [15]. Distinct nucleotides are detected only with their bases. Those bases are, respectively, abbreviated as adenine (A) , guanine (G), cytosine (C), and thymine (T). Two strands of DNA can form (under appropriate conditions) a double strand, if the respective bases are the Watson-Crick complements of each other: A matches T and C matches G; also $3'$ end matches $5'$ end. The length of a single stranded DNA is the number of nucleotides comprising the single strand. Thus, if a single stranded DNA includes 20 nucleotides, it is called a 20*mer*. The length of a double stranded DNA (where each nucleotide is base paired) is counted in the number of base pairs. Thus, if we make a double stranded DNA from a single stranded 20 mer, then the length of the double stranded DNA is 20 base pairs, also written as 20 bp.

The DNA operations proposed by Aldeman [2] and Lipton [3] are described below. These operations will be used for figuring out solutions of the minimum vertex cover problem in this paper. The Adleman-Lipton model: A (test) tube is a set of molecules of DNA (i.e., a multi-set of finite strings over the alphabet {A,C,G,T}). Given a tube, one can perform the following operations:

(1) *Merge* (T_1,T_2): for two given test tubes T_1,T_2, it stores the union $T_1\bigcup T_2$ in T_1 and leaves T_2 empty;

(2) *Copy* (T_1,T_2): for a given test tube T_1, it produces a test tube T_2 with the same contents as T_1 ;

(3) *Detect (T)*: given a test tube T, it outputs "yes" if T contains at least one strand, otherwise, outputs "no";

(4) *Separation* (T_1,X,T_2): for a given test tube T_1 and a given set of strings X, it removes all single strands containing a string in X from T_1, and produces a test tube T_2 with the removed strands;

(5) *Selection* (T_1,L,T_2): for a given test tube T_1 and a given integer L, it removes all strands with length L from T_1, and produces a test tube T_2 with the removed strands;

(6) *Discard (T)*: for a given test tube T, it discards the tube T;

(7) *Read (T)*: for a given tube T, the operation is used to describe a single molecule, which is contained in the tube T. Even if T contains many different molecules each

encoding a different set of bases, the operation can give an explicit description of exactly one of them;

(8)*Append-tail (T,Z)*: for a given test tube T and a given DNA singled strand Z, it appends Z onto the end of every strand in the tube T.

Since these eight manipulations are implemented with a constant number of biological steps for DNA strands [14], we assume that the complexity of each manipulation is $O(1)$ steps.

3 DNA algorithm for the minimum edge cover problem

For a given undirected and simple graph $G=(V,E)$, $V=\{v_k|k=1,2,\ldots,n\}$ is vertex set and $E=\{e_{i,j}|1\leq i<j\leq n\}$ is edge set. Some vertices v_i and v_j can be connected by the edge $e_{i,j}$ $(i<j)$ in graph. We let $|E|=m$ and $m\leq n(n+1)/2$. At the same time, the simple graph processed in this paper has no self-loops.

In the following, the symbols $\#, X, 0, 1, A_k$ $(k=1,2,\ldots,n)$ denote distinct DNA singled strands with same length, say t-mer (t can choose a small integer, such as 5 mer). Obviously the length t of the DNA singled strands greatly depends on the size of the problem involved in order to distinguish all above symbols and to avoid hairpin formation. Then in the below operations, we use the distinct DNA singled strands symbols A_i0A_j, A_i1A_j $(1\leq i<j\leq n)$ to denote the edge $e_{i,j}$, with A_i1A_j for in the edge subset, while A_i0A_j for not. For distinguishing some edges in a edge subset or not, we meantime design DNA string X with t-mer length. the symbols # means starting signal of the strands. Let

$$T_0=\{\#\},$$

For a graph with n vertices and m edges, every possible subset of the edge subset E' can be expressed by an m-digit binary number. A bit set to 1 represents the edge in the subset, and a bit set to 0 represents the edge out of the subset. For example in Fig. 1, the subset $\{e_{1,3},e_{2,3},e_{3,4},e_{4,6}\}$ can be expressed by the binary number 01101001. In this way, we transform all possible subsets of E in a m-edge graph into an ensemble of all m-digit binary numbers. We call this the data pool.

(1)We get all possible subsets of edge in graph.

For $i=1$ to $i=n-1$
  For $j=i$ to $j=n$
(1-1)If$(e_{i,j}\in E)$
  Then
(1-2)$Copy(T_0,T_1)$;
(1-3)$Copy(T_0,T_2)$;
(1-4)$Discard(T_0)$;
(1-5)$Append-tail(T_1,A_i1A_j)$;
(1-6)$Append-tail(T_2,A_i0A_j)$;
(1-7)$Merge(T_0,T_1)$;
(1-8)$Merge(T_0,T_2)$;
  End for
End for

After the above steps of manipulations, the singled strands in tube T_0 will encode all possible subsets of edge. For example, for the graph in Fig. 1, we have singled strands:

$$\#A_11A_2A_10A_3A_21A_3A_21A_4A_30A_4$$

$$A_31A_5A_40A_5A_41A_6\in T_0$$

which denotes the subset of edge

$$\{e_{1,2},e_{2,3},e_{2,4},e_{3,5},e_{4,6}\}$$

corresponding to the binary number 10110101. Meanwhile we use two "For" clauses, thus these operations can be finished in $O(n^2)$ steps since each single manipulation above works in $O(1)$ steps.

(2)Each singled strand in tube T_0 denotes one possible edge subset. The minimum edge cover problem is firstly require that any vertex of the graph at least belongs to one edge of the subset. So we should check all the edge subsets whether to satisfy the above condition. We should discard the strands which vertex $v_i(1\leq i\leq n)$ is not in the subset. For example in Fig. 1, the singled strands

$$\#A_10A_2A_11A_3A_20A_3A_20A_4A_31A_4$$

$$A_30A_5A_41A_5A_41A_6\in T_0$$

(representing the subset of edge $\{e_{1,3},e_{3,4},e_{4,5},e_{5,6}\}$) should be discarded for the vertex v_2 is not in the subset. We can choose all possible edge cover subsets in graph through following algorithm.

For $i=1$ to $i=n$
(2-1)$Separation(T_0,A_i1,T_3)$;
(2-2)$Separation(T_0,1A_i,T_4)$;
(2-3)$Discard(T_0)$;
(2-4)$Merge(T_3,T_4)$;
(2-5)$Copy(T_3,T_0)$;
(2-6)$Discard(T_3)$;
End for

After the above operations, the singled strands in tube T_0 are all edge cover subsets. Meanwhile we use one "For" clauses, thus this operation can be finished in $O(n)$ steps since each single manipulation above works in $O(1)$ steps.

(3)The minimum edge cover problem should be a smallest edge cover subset which satisfy above condition in graph. So we choose the leastest edge subset in all edge cover subsets. If a edge $e_{i,j}$ in the edge subset, we append additional strand X at the end of previous strand in order to find the optimum strand solutions. For example, for the graph in Fig. 1, the singled strands

$$\#A_10A_2A_11A_3A_20A_3A_20A_4A_31A_4$$

$$A_30A_5A_41A_5A_41A_6 \in T_0$$

represent containing the edges $\{e_{1,3}, e_{3,4}, e_{4,5}, e_{4,6}\}$, So we append strand X four times at the end of previous strands to

$$\#A_10A_2A_11A_3A_20A_3A_20A_4A_31A_4$$

$$A_30A_5A_41A_5A_41A_6XXXX$$

This is done by the following manipulations:
For $i = 1$ to $i = n-1$
 For $j = i+1$ to $j = n$
(3-1)$Separation(T_0, A_i1A_j, T_5)$;
(3-2)$Append-tail(T_5, X)$;
(3-3)$Merge(T_0, T_5)$;
 End for
End for
In the above operation, we use two "For" clause, thus this operation can be finished in $O(n^2)$ steps since each single manipulation above works in $O(1)$ steps.

(4)We take out those singled strands in T_0 with smallest length, which give the solutions to minimum edge cover problem. For example, for the graph in Fig. 1, those singled strands in T_0 with leastest length are

$$\#A_11A_2A_10A_3A_20A_3A_20A_4A_30A_4$$

$$A_31A_5A_40A_5A_41A_6XXX \in T_0.$$

Therefore, solutions to minimum edge cover problem in Fig. 1 is the edge subset with $\{e_{1,2}, e_{3,5}, e_{4,6}\}$.
For $i = 1$ to $i = m$
(4-1)$Selection(T_0, (3m+1+i)t, T_6)$;
(4-2)If($Detect(T_6)$ = "Yes")
 Then
 break;
End for
(4-3)$Read(T_6)$;
In the above operation, we use one "For" clause, the worst conditions is that the algorithm stop at $i = m$ and $m \le n(n+1)/2$, thus this operation can be finished less than $O(n^2)$ steps since each single manipulation above works in $O(1)$ steps. Finally the "*Read*" operation is applied to giving the exact solutions of the minimum edge cover problem.

4 The complexity and feasibility of the proposed DNA algorithm

The following theorem tells that the algorithm proposed above really can get solutions of the minimum edge cover problem in $O(n^2)$ steps using DNA molecules.

Theorem 1. *The solutions of minimum edge cover problems for a graph with n vertices and m edges can be solved by the above DNA operations.*

Proof. We first get all possible combinations of the edges in the data pool after the first step. Because any vertex in graph should at least belong to one edge of the subset, basic biological operations are used to remove illegal solution and find legal solution from data pool through at the second step. In order to choose the minimum edge subset, at the third step we append a series of "tails" X at the end of the strands if some edges belong to the edge subset. The shortest strands in the pool mean the solution of minimum edge cover problem, and we can "read" the answer at the last step.

Theorem 2. *The solutions of minimum edge cover problems for a graph with n vertices and m edges can be figured out in $O(n^2)$ steps using DNA molecules.*

Proof. The manipulates of algorithm can be entirely finished in finite operations. Such as step (1) and step (3) in $O(n^2)$, step (4) less than $O(n^2)$, Simultaneity step (2) in $O(n)$. In conclusion, We can get the solution of minimum edge cover problems with n vertices and m edges in $O(n^2)$.

Theorem 3. *The solutions of minimum edge cover problems for a graph with n vertices and m edges can be founded between $(3m+2)t$ and $(4m+1)t$ length range.*

Proof. After the operations of first step, all the singled strands in tube T_0 denote all possible edge subsets. Then strands can be described:

$$\#A_1y_{1,k}A_k \cdots A_iy_{i,j}A_j \cdots A_ly_{l,n}A_n \quad y_{i,j} = 0 \ \ or \ \ 1$$

After the operations of second step, all the strands in T_0 contain all the vertices information in the edge subsets. We reasonably design the length of $\#, A_k, y_{i,j}$ and X, For

$$||A_k|| = ||\#|| = ||y_{i,j}|| = ||X|| = t$$

So we define S as the strands after the third step. Then S can be described:

$$\#A_1y_{1,k}A_k \cdots A_iy_{i,j}A_j \cdots A_ly_{l,n}A_nX \cdots X$$

The number p of appending X times is decided by the existing edges information on the strands. Due to the possible of containing edges $e_{i,j}$ information between 1 and m in the edge subset, So

$$\begin{aligned}
&||S|| = ||\#|| + ||A_1|| + ||y_{1,k}|| + ||A_k|| + \cdots + ||A_l|| \\
&+||y_{l,n}|| + ||A_n|| + ||X|| + \cdots + ||X|| \\
&= ||\#|| + 2\sum_{i=1}^{m}||A_i|| + \sum_{i=1}^{m}||y_{i,j}|| + p||X|| \\
&= (3m+1)t + pt \\
&\because 1 \le p \le m \\
&\therefore (3m+2)t \le ||S|| \le (4m+1)t
\end{aligned}$$

So the length of strands which denote containing all the vertices information must be between $(3m+2)t$ and $(4m+1)t$. So we can get the solution in step 4 in appropriate length range.

5 Conclusion

In this paper, we present DNA algorithms for solving the minimum edge cover problem based on biological operations in the Adleman-Lipton model. The proposed algorithms have two advantages. Firstly, the proposed algorithm actually has a lower rate of errors for hybridization because we generate fixed reasonable DNA sequences for generating the solutions of the problem. Secondly, the proposed algorithms can works in $O(n^2)$ steps for the minimum edge cover problem of an undirected and simple graph with n vertices and m edges, Comparing exponential level time by electronic computer. The ability to perform complex operations in solution might help us learn more about the nature of computation and lead to the development of better DNA based computation, capable of solving a wide range of complex problems.

Acknowledgement

The first author acknowledges the financial support by Doctor fund of Shanghai Ocean University (A-2400-12-0000351) and Shanghai Ocean university 085 Engineering. The second author acknowledges the financial support by CNSF (Grant numbers: 61272098).The corresponding author (Chengpei Tang) acknowledges the financial support by Guangdong Province special funds for the development of strategic emerging industry(2012556036).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] Garey, M.R., Johnson, D.S., Computers and Intractability: A Guide to the Theory of NP-completeness. (W. H. Freeman and Company, 1979).
[2] L.M. Adleman, Molecular computation of solution to combinatorial problems, Science **266**, 1021-1024 (1994).
[3] R.J. Lipton, DNA solution of HARD computational problems, Science **268**, 542-545 (1995).
[4] A. Fujiwara, K. Matsumoto, Wei Chen, Procedures for logic and arithmetic operations with DNA molecules, International Journal of Foundations of Computer Science **15**, 461-474 (2004).
[5] F. Guarnieri, M. Fliss, C. Bancroft, Making DNA add, Science **273**, 220-223 (1996).
[6] H. Hug, R. Schuler, DNA-based parallel computation of simple arithmetic, in: Proceedings of the 7th International Meeting on DNA Based Computers 159-166 (2001).
[7] S. Kamio, A. Takehara, A. Fujiwara, Procedures for computing the maximum with DNA strands, in: Humid R. Arabnia, Youngsong Mun (Eds.), Proceedings of the International Conference on DNA Based Computers (2003).
[8] W.X. Li, D.M. Xiao, L. He, DNA ternary addition, Applied Mathematics and Computation **182**, 977-986 (2006).
[9] D.M. Xiao, W.X. Li, J. Yu, X.D. Zhang, Z.Z. Zhang, L. He, Procedures for a dynamical system on $\{0,1\}^n$ with DNA molecules, BioSystems **84**, 207-216 (2006).
[10] X.L. Wang, Z.M. Bao, J.J. Hu, S. Wang, A. Zhan, Soling the SAT problem using a DNA computing algorithm based on ligase chain reaction, BioSystems **91**, 117-125 (2008).
[11] J.-Y. Lee, S.-Y. Shin, T.-H. Park, B.-T. Zhang, Solving traveling salesman problems with DNA molecules encoding numerical values, BioSystems **78**, 39-47 (2004).
[12] A. Narayanan, S. Zorbalas, et al., DNA algorithms for computing shortest paths, in: J.R. Koza (Ed.), Proceedings of the Genetic Programming 1998, Morgan Kaufmann, 718-723 (1998).
[13] Xuncai Zhang, Yanfeng Wang,Guangzhao Cui, et al., Application of a novel IWO to the design of encoding sequences for DNA computing, Computers and Mathematics with Applications **57**, 2001-2008 (2009).
[14] M. Yamamura, Y. Hiroto, T. Matoba, Solutions of shortest path problems by concentration control, Lecture Notes Computer Science **2340**, 231-240 (2002).
[15] Wen Li, Lu-Bin Cui, Michael K. Ng, On computation of the steady-state probability distribution of probabilistic Boolean networks with gene perturbation,Journal of Computational and Applied Mathematics **236**, 4067-4081 (2012).
[16] Jianhua Xiao, Jin Xu, Zhihua Chen, et al., A hybrid quantum chaotic swarm evolutionary algorithm for DNA encoding, Computers and Mathematics with Applications **57**, 1949-1958 (2009).
[17] Li, D., Huang, H., Li, X., Li, X. Hairpin formation in DNA computation presents limits for large NP-complete problems. BioSystems **72**, 203-207 (2003).
[18] Q. Ouyang, Peter D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem, Science **278**, 446-449 (1997).
[19] M.Y. Guo, W.L. Chang, M. Ho, J. Lu, J.N. Cao, Is optimal solution of every NP-complete or NP-hard problem determined from its characteristic for DNA-based computing, BioSystems **80**, 71-82 (2005).
[20] Z.C. Wang, D.M. Xiao, W.X. Li, L. He, A DNA procedure for solving the shortest path problem, Applied Mathematics and Computation **183**, 79-84 (2006).
[21] Zhang, Qiang; Zhou, Shihua; Wei, Xiaopeng,An Efficient Approach for DNA Fractal-based Image Encryption, Appl. Math. Inf. Sci. **5**, 445-459 (2011).
[22] Z.C. Wang, Yiming Zhang, Weihua Zhou, Haifeng Liu, Solving traveling salesman problem in the Adleman-Lipton model, Applied Mathematics and Computation **219**, 2267-2270 (2012).
[23] Lee, Cheng-Chi; Chen, Shun-Der; Chen, Chin-Ling, A Computation-Efficient Three-Party Encrypted Key Exchange Protocol, Appl. Math. Inf. Sci. **6**, 573-579 (2012).

[24] Hsieh, Yi-Chih; You, Peng-Sheng ,A New Space-Filling Curve Based Method for the Traveling Salesman Problems, Appl. Math. Inf. Sci. **6**, 371-377 (2012).

Zhaocai Wang was born in Shandong, China in 1979. He received the BS degree in Applied Mathematics from Shanghai Jiaotong University and the PhD degree from Fudan University in 2006 and 2012 resperctively. Since 2006 he is a lecturer in information school at Shanghai Ocean university. His research interests include biology mathematics and biology computation. Dr Wang has published over 15 papers in leading journals and conferences.

Dongmei Huang was born in Henan, China in 1964. She received the BS degree in Computer and Automation from the Zhengzhou University in 1990. Since 2000 she is a lecturer in information school at Shanghai Ocean university and since 2005 she is the information science group leader. Her research interests include chaos of nonlinear system and electrical engineering. Mrs Huang has published over 60 papers in leading journals and conferences.

Chengpei Tang received the PhD degree in Radio Physics from the department of Electronics and Communication Engineering, Sun Yat-sen University in 2007. His research field are network architecture, routing systems.