

Molecular solutions for the minimum edge dominating set problem on DNA-based supercomputing

Saeed Safaei¹, Babak Dalvand², Babak Esmaeili³, and Vahid Safaei⁴

¹Department of Mathematics, Arak University, Arak, Markazi, Iran

²Department of Mathematics, Islamic Azad University of Khoramabad, Lorestan, Iran

³Department of Computer Engineering, Islamic Azad University of Najafabad, Isfahan, Isfahan, Iran

⁴Department of Mechanical Engineering, Islamic Azad University of Khomeini Shahr, Arak, Markazi, Iran

Abstract - Adleman showed that deoxyribonucleic acid (DNA) strands could be employed towards calculating solutions to an instance of the Hamiltonian path problem (HPP). Lipton also demonstrated that Adleman's techniques could be used to solve the Satisfiability problem. In this paper, we use Adleman-Lipton model for developing a DNA algorithm to solve minimum edge dominating set problem (MEDSP). In spite of the NP-hardness of minimum edge dominating set problem (MEDSP) our DNA procedures is done in a polynomial time.

Keywords: DNA computing, minimum edge dominating set problem.

1 Introduction

Recently, DNA computing has considerable attention as one of non-silicon based computing. Watson-Crick complementarity and massive parallelism are two important features of DNA. By using these features, one can solve an NP-complete problem, which usually needs exponential time on a silicon-based computer, in a polynomial number of steps with DNA molecules. Adleman [1] solved Hamiltonian path problem of size n in spite of NP-hardness of the problem in $O(n)$ steps using DNA molecules. That is the first work for DNA computing. The second NP-hard problem that has solved by DNA computing is Satisfiability (SAT), Lipton [12] showed that the Adleman's manner could be used to determine SAT. Ouyang et al. [14] used the model to solve maximal clique problem. Some other NP-hard problems that have been solved by the model are as follow: binary integer programming [20], exact cover by 3-sets [4], maximum cut [19] etc. Moreover, procedures for primitive operations, such as logic or arithmetic operations, have also been proposed so as to apply DNA computing in a wide range of problems [2,5-7, 9-11, 16].

In this paper, the DNA operations proposed by Adleman (1994) and Lipton (1995) are used for figuring out solutions of minimum edge dominating set problem.

For a given Graph $G=(V,E)$ we want to find an edge dominating set for G , i.e., a subset $E' \subset E$ such that for all $e_1 \in E - E'$ there is an $e_2 \in E'$ such that e_1 and e_2 are adjacent and the size of $|E'|$ is minimum.

The rest of this paper is organized as follows. In Section 2, the Adleman-Lipton model is introduced in detail. Section 3 we present a DNA algorithm for solving the minimum edge dominating set problem and the complexity of the proposed algorithm is described. We give conclusions in Section 4.

2 Adleman-Lipton model

Bio-molecular computers work at the molecular level. Since biological and mathematical operations have some similarities, DNA, the genetic material that encodes the living organisms, is stable and predictable in its reactions and can be used to encode information for mathematical problems. DNA algorithms typically solve problems by initially assembling large data sets as input and then eliminating undesirable solutions.

A DNA (deoxyribonucleic acid) is a polymer, which is strung together from monomers called deoxyribonucleotides [15, 17]. Distinct nucleotides are detected only with their bases.

Those bases are 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, i.e., A matches T and C matches G; also 3'- end matches 5'- end. For example, strands 5'-ACCGGATGTCA-3' and 3'-TGGCCTACAGT-5' can form a double strand. We also call them as the complementary strand of each other.

The length of a single DNA strand is the number of nucleotides comprising the single strand. Thus, if a single DNA strand includes 20 nucleotides, it is called a 20 mer. The length of a double strand (where each nucleotide is base paired) is counted in the number of base pairs. Thus, if we make a double strand from two single strands of length 20

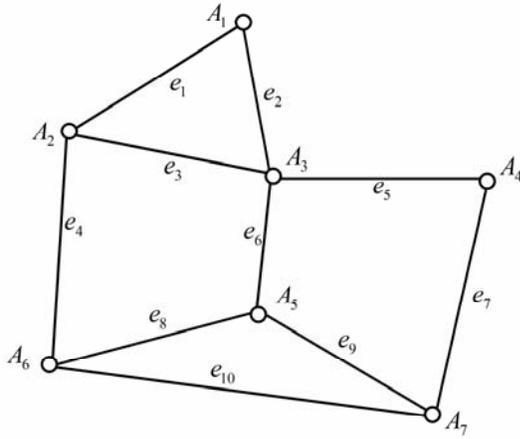


Fig. 1. Graph G.

mer, then the length of the double strand is 20 base pairs, also written as 20 bp for more discussion of the relevant biological background, refer to [3, 15, and 17]. The DNA operations proposed by Adleman and Lipton [1, 3, and 12] are described below.

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}). The following operations perform on tubes:

- (1) *Merge* (T_1, T_2): for two given test tubes T_1, T_2 it stores the union $T_1 \cup T_2$ 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) *Cleavage* ($T, \sigma_0\sigma_1$): for a given test tube T and a string of two (specified) symbols $\sigma_0\sigma_1$ it cuts each double strand containing $\begin{bmatrix} \sigma_0\sigma_1 \\ \sigma_0\sigma_1 \end{bmatrix}$ in T into two double strands as follows:

$$\begin{bmatrix} \alpha_0\sigma_0\sigma_1\beta_0 \\ \alpha_1\sigma_0\sigma_1\beta_1 \end{bmatrix} \Rightarrow \begin{bmatrix} \alpha_0\sigma_0 \\ \alpha_1\sigma_0 \end{bmatrix}, \begin{bmatrix} \sigma_1\beta_0 \\ \sigma_1\beta_1 \end{bmatrix}$$

- (7) *Annealing* (T): for a given test tube T it produces all feasible double strands in T . The produced double strands are still stored in T after Annealing;
- (8) *Denaturation* (T): for a given test tube T it dissociates each double strand in T into two single strands;
- (9) *Discard* (T): for a given test tube T it discards the tube T ;
- (10) *Append* (T, Z): for a given test tube T and a given short DNA singled strand Z it appends Z onto the end of every strand in the tube T ;
- (11) *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.

Since these eleven manipulations are implemented with a constant number of biological steps for DNA strands (Păun et al., 1998), we assume that the complexity of each manipulation is $O(1)$ steps.

3 Solving MEDSP by Adleman-Lipton Model

Let $G = (V, E)$ be an undirected graph with the set of vertices being $V = \{A_k \mid k = 1, 2, \dots, m\}$ and the set of edges being $E = \{e_i \mid i = 1, 2, \dots, n\}$. Let $|E|=d$. In the following, the symbols $0, 1, 2, \#, X, Y, A_k, B_j, C_j$ ($k = 1, 2, \dots, m, j = 1, 2, \dots, m$) denote distinct DNA singled strands with same length, say 10-mer. And $\|\cdot\|$ denotes the length of the DNA singled strand. Obviously the length 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 (Li et al., 2003).

Tubes P and Q are defined as follows:

Let

$$P = \{ \#B_nC_n, B_kC_k, X, \#, 0, 1, B_j, A_j \mid k = 1, 2, \dots, n-1, j = 1, 2, \dots, m \}$$

$$Q = \{ \#, C_i0A_j0A_kB_{i-1}, C_i1A_j1A_kB_{i-1}, C_11A_j1A_k\#, \\ \#, C_10A_j0A_kX\# \mid k = 1, 2, \dots, n, j = 1, 2, \dots, n \}$$

We design the following algorithm to solve the minimum edge dominating set problem and give the corresponding DNA operations as follows:

3.1 Produce each possible subset from E

For a set E with n edges, each possible *subset* of edges is represented by an n -digit number. For example for graph 1 we can represent $E_1 = \{e_7, e_4, e_2, e_1\}$ as 0001001011 and show $E_2 = \{e_8, e_7, e_5, e_4, e_1\}$ as 0011011001, in which number 1 in i -th element shows that the edges e_i is in the subset, and 0 it means that this edges doesn't exist in the subset.

Hence, we show all subsets of E with an n -digit number. Each DNA strand represents a subset of E and the strand $B_i C_i 1 A_j 1 A_k$ in the DNA strand illustrates that the i -th edge exists in the substring equal to this strand and the strand $B_i C_i 0 A_j 0 A_k$ shows that the same edge doesn't exist in the so mentioned substring.

A_j, A_k are the vertices which the i -th edge has connected together.

We call this the data pool.

- (1-1) Merge (P, Q);
- (1-2) Annealing (P);
- (1-3) Denaturation (P);
- (1-4) Separation ($P, \{X\# \}, T_{mp}$);
- (1-5) Discard (P);
- (1-6) Separation ($T_{mp}, \{\#B_n \}, P$);

After above six steps of manipulation, singled strands in tube P will encode all 2^n subsets from E in the form of n -digit binary numbers. For example, for the graph in Fig. 1 with $n=7$ we have, e.g. the singled strand
 $\#B_{10}C_{10}1A_61A_7B_9C_91A_51A_7B_8C_80A_50A_6B_7C_70A_40A_7$
 $B_6C_60A_30A_5B_5C_51A_31A_4B_4C_41A_21A_3B_3C_30A_20A_3$
 $B_2C_20A_10A_3B_1C_11A_11A_2X\#$

Which denotes the subset $E' = \{e_{10}, e_9, e_5, e_4, e_1\}$ corresponding to the number 1100011001. This operation can be finished in $O(1)$ steps since each manipulation above works in $O(1)$ steps.

3.2 Eliminating the sets not having the condition

In this phase we create an array called S with dimensions equal to $2*n$. The indexes of the two vertices which

encompass edge e_j are put in the i -th row as shown in figure 2. At this stage, first of all, all subsets not containing edge e_n are found then in the found sets we look for all subsets which do not have e_n, e_l adjacently. We take away these sets and add Y sub-strand to the end of the set. As can be clearly seen these sets are impossible. Finally, any strands containing Y sub-strand are eliminated. Same is done for all the other edges of $e_{n-1}, e_{n-2}, \dots, e_1$. We clearly know that if edge e_i is

Edge	Vertex	Vertex
e_1	A_1	A_2
e_2	A_1	A_3
e_3	A_2	A_3
e_4	A_2	A_6
e_5	A_3	A_4
e_6	A_3	A_5
e_7	A_4	A_7
e_8	A_5	A_6
e_9	A_5	A_7
e_{10}	A_6	A_7

Fig. 2

Encompassed in between u, v then, each edge is adjacent to e_i which has u and v at .

For $r = 1$ to $r = n$

- (2-1) Separation ($P, \{B_r C_r 0 \}, T_1$)
- (2-2) Separation ($T_1, 1S[r][1], T_2$)
- (2-3) Separation ($T_1, 1S[r][2], T_3$)
- (2-4) Merge (T_2, T_3)
- (2-5) Append (T_1, Y)
- (2-6) Merge (P, T_1)
- (2-7) Merge (P, T_2)
- (2-8) Discard (T_1)

End for

- (2-9) Separation (P, Y, T_1)
- (2-10) Discard (T_1)

Time analysis of the above algorithm

Each of the above actions will conclude at $O(1)$. Therefore the algorithm will terminate at $O(n)$.

3.3 Finding the subset(s) with most members amongst

At this stage, firstly edges of each subset are calculated and then the subset with the least edges is found. If a strand contains edge e_j , X is added to the end of that string. The length of each $B_i C_i 1 A_g 1 A_k$ or $B_i C_i 0 A_g 0 A_k$ strand equals to 60 and the length of $\#, X \#$ strands equals to 30 so the length of each strand represents a subset equal to $60 * n + 30$. The following algorithm finds the strand which has the minimum number of members.

For $r = 1$ to $r = n$

(3-1) Selection($P, 60 * n + 30 + (n - r) * 10, T_1$)

(3-2) If Detect(T) is yes,

then end for else continue the circulation

End for

Time analysis of the above algorithm

Each of the above actions will conclude at $O(1)$. Therefore the algorithm will terminate at $O(n)$.

4 Conclusions

As the first work for DNA computing, (Adleman, 1994) presented an idea to demonstrate that deoxyribonucleic acid (DNA) strands can be applied to solving the Hamiltonian path NP-complete problem of size n in $O(n)$ steps using DNA molecules. Adleman's work shows that one can solve an NP-complete problem, which usually needs exponential time on a silicon-based computer, in a polynomial number of steps with DNA molecules. From then on, Lipton (1995) demonstrated that Adleman's experiment could be used to determine the NP-complete Satisfiability (SAT) problem (the first NP-complete problem). Ouyang et al. (1997) showed that restriction enzymes could be used to solve the NP-complete clique problem. In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems. As Guo et al. (2005) pointed out, it is still important to design DNA procedures and algorithms for solving various NP-complete problems since it is very difficult to use biological operations for replacing mathematical operations.

In this paper, we propose a procedure for minimum edge dominating set NP-complete problems in the Adleman-Lipton model. The procedure works in $O(n^3)$ steps for minimum edge dominating set problem of a directed graph with n vertices. All our results in this paper are based on a theoretical model. However, the proposed procedures can be implemented practically since every DNA manipulation used in this model has been already realized in lab level.

5 References

- [1] Adleman L.M., Molecular computation of solution to combinatorial problems, Science 266 pp 1021-1024 (1994).
- [2] Ardashir Dolati., M. S. Haghghat., Saeed Safaei., and Hajar Mozaffar., Solving Minimum B-Vertex Separator Problems in the Adleman-Lipton Model. FCS'08, USA pp 97-101 (2008)
- [3] Boneh, D., Dunworth, C., Lipton, R.J., Sgall, J., On the computational power of DNA. Discrete Applied Mathematics. 71 (1996) pp. 79-94 (1996).
- [4] Chang W.L., Guo M., Solving the set cover problem and the problem of exact cover by 3-sets in the Adleman-Lipton model. BioSystems 72 pp 263-275 (2003).
- [5] Frisco, P., Parallel arithmetic with splicing. Romanian J. Inf. Sci. Technol. 2 pp 113-128 (2002).
- [6] Fujiwara, A., Matsumoto, K., Chen, Wei, Procedures for logic and arithmetic operations with DNA molecules. Int. J. Found. Comput. Sci. 15 pp 461-474 (2004).
- [7] Guarnieri, F., Fliss, M., Bancroft, C., Making DNA add. Science 273 pp 220-223 (1996).
- [8] Guo M.Y., Chang W.L., Ho M., Lu J., Cao J.N., Is optimal solution of every NP-complete or NP-hard problem determined from its characteristic for DNA-based computing, BioSystems 80 pp 71-82 (2005).
- [9] Gupta, V., Parthasarathy, S., Zaki, M.J., Arithmetic and logic operations with DNA. In: Proceedings of Third DIMACS Workshop on DNA-Based Computers, pp 212-220 (1997).
- [10] Hug, H., Schuler, R., DNA-based parallel computation of simple arithmetic. In: Proceedings of the Seventh International Meeting on DNA-based Computers, pp 159-166 (2001).
- [11] Kamio, S., Takehara, A., Fujiwara, A., Procedures for computing the maximum with DNA strands. in: Arabnia, Hamid, R., Mun, Youngsong (Eds.), In: Proceedings of the International Conference on DNA-Based Computers. (2003).
- [12] Lipton R.J., DNA solution of HARD computational problems, Science 268 pp 542-545 (1995).
- [13] Lee J.-Y., Shin S.-Y., Park T.-H., Zhang B.-T., Solving traveling salesman problems with DNA molecules encoding numerical values, BioSystems 78 pp 39-47 (2004).

- [14] Ouyang Q., Kaplan P. D., Liu S., Libchaber A., DNA solution of the maximal clique problem, *Science* 278 pp 446-449 (1997).
- [15] Paun, G., Rozenberg, G., Salomaa, A., *DNA Computing: New Computing Paradigms*. Springer, New York. 1998.
- [16] Saeed Safaei., Hajar Mozaffar., and Babak Esmaili., Solving Minimum K-Center Problem in the Adleman-Lipton Model. FCS'08, USA pp 182-186 (2008)
- [17] Sinden, R.R., *DNA Structure and Function*. Academic Press, New York. 1994.
- [18] Wang Z., Xiao D., Li W., He L., A DNA procedure for solving the shortest path problem, *Applied Mathematics and Computation* 183 pp 79-84 (2006).
- [19] Xiao D., Li W., Zhang Z., He L., Solving maximum cut problems in the Adleman-Lipton model. *BioSystems* 82 pp 203-207 (2005).
- [20] Yeh C.W., Chu C.P., Wu K.R., Molecular solutions to the binary integer programming problem based on DNA computation *BioSystems* 83 pp 56-66 (2006).