Solving minimum β-vertex separator problems in the Adleman–Lipton model

Ardashir Dolati[1], M.S Haghighat[2], Saeed Safaei[2,*], Hajar Mozaffar[3]
1) Department of Mathematics, Shahed University, Tehran, Tehran, Iran
2) Department of Mathematics, Arak University, Arak, Markazi, Iran
3) Computer Engineering, Isfahan University, Isfahan, Isfahan, Iran

Abstract – *In recent works for high-performance computing, computation with DNA molecules, i.e. DNA computing, has considerable attention as one of non-silicon-based computing. Watson–Crick complementarity and massive parallelism are two important features of DNA. Using the 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. In this paper, we consider a procedure for solving minimum β-vertex separator problems in the Adleman–Lipton model. The procedure works in $O(n^2)$ steps for minimum β-vertex separator problems of an undirected graph with n vertices.*

Keywords: minimum β-vertex separator problem; Adleman–Lipton model;

1 Introduction

As the first work for DNA computing, Adleman (1994) presented an idea of solving the Hamiltonian path problem of size n in *O(n)* steps using DNA molecules. 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) presented a molecule biology-based experimental solution to the maximal clique NP-complete problem. In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems. Moreover, procedures for primitive operations, such as logic or arithmetic operations, have been also proposed so as to apply DNA computing on a wide range of problems (Frisco, 2002; Fujiwara et al., 2004; Guarnieri et al., 1996; Gupta et al., 1997; Hug and Schuler, 2001; and Kamio et al., 2003).

In this paper, the DNA operations proposed by Adleman (1994) and Lipton (1995) are used for figuring out solutions of *minimum β-vertex separator NP-complete problems*: for a graph $G=(V,E)$ and a rational β, $0 \le \beta \le \frac{1}{2}$. Find a partition of V into disjoint sets R, S and T such that $\max\{|R|,|S|\} \le \beta^*|V|$ and no edge has one endpoint in R and one in S and also the size of $|T|$ is minimized.

The graph G in Fig. 1 defines such a problem. It is easy to see that $R=\{A_4,A_5,A_6\}, S=\{A_2,A_3\}, T=\{A_1,A_7,A_8\}$ is a solution to the minimum β-vertex separator problem with $\beta=\frac{3}{7}$ 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 we present a DNA algorithm for solving the minimum β-vertex separator problem and the complexity of the proposed algorithm is described. We give conclusions in Section 4.

2 The Adleman–Lipton model

Bio-molecular computers work at the molecular level. Because biological and mathematical operations have some similarities, DNA, the genetic material that encodes for living organisms, is stable and predictable in its reactions and can be used to encode information for mathematical systems.

A DNA (deoxyribonucleic acid) is a polymer which is strung together from monomers called deoxyribo-nucleotides (Pâun et al., 1998). Distinct nucleotides are detected only with their bases. Those bases are, respectively, abbreviated as A (adenine), G (guanine), C (cytosine) and T (thymine). 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, e.g. the singled strands 5'-ACCGGATGTCA-3' and 3' –TGGCCTACAGT-5' can form a double strand. We also call the strand 3'-TGGCCTACAGT-5' as the complementary strand of 5'-ACCGGATGTCA-3' and simply denote 3'-TGGCCTACAGT-5' by $\overline{\text{ACCGGATGTCA}}$. 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

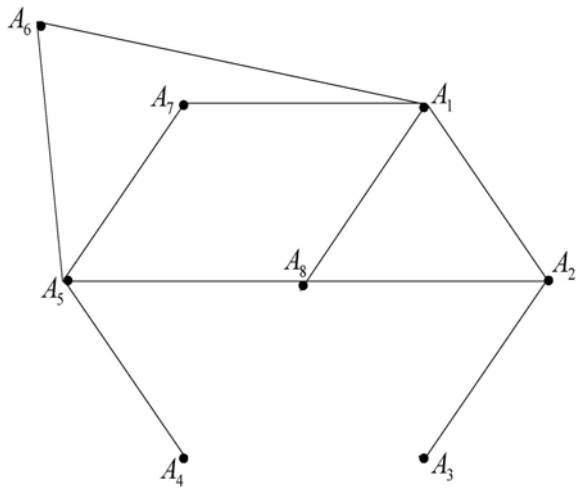

Fig. 1. Graph G.

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 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 $T_1 \cup T_2$ 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 trend containing $\begin{bmatrix}\sigma_0\sigma_1\\ \sigma_0\sigma_1\end{bmatrix}$ in T into two double strands as follows:

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

(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 (Pâun et al., 1998), we assume that the complexity of each manipulation is $O(1)$ steps.

3 DNA algorithm for the minimum β-vertex separator problems

Let $G=(V,E)$ be a graph with the set of vertices being $V=\{A_k \mid k=1,2,\ldots,n\}$ and the set of edges being $E=\{e_{ij} \mid for\ some\ 1\le i,j\le n\}$. Let |E|=d. Then $d\le\frac{n(n-1)}{2}$. Note that e_{ij} is in E if the vertices A_i and A_j are connected by an edge.

In the following, the symbols $0,1,2,3,4,5,\#,A_k,B_k\ (k=1,2,\ldots,n)$ denote distinct DNA singled strands with same length, say 10-mer. 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). We choose DNA singled strands y_{ij} to encode the edges connecting the vertices A_i and A_j with length of 10-mer. Here, all these y_{ij} will be taken the same, say y_{11}, for our problem. For convenience of argument we still use a dummy symbol y_{ij} of length 0-mer if the vertices A_i and A_j is not connected by an edge or $i=j$. Let

$P = \{0,1,2,3,4,5, A_1\#, \#B_n, A_k B_{k-1} \mid k = 2,3,\ldots,n\}$,
$Q = \{\overline{\#}, \overline{B_k 2A_k}, \overline{B_k 1A_k}, \overline{B_k 0A_k} \mid k = 1,2,\ldots,n\}$, *and*
$H = \{y_{ij} \mid i, j = 1,2,\ldots,n\}$.

We design the following algorithm to solve the minimum β-vertex separator problems and give the corresponding DNA operations as follows:

3.1 Represent each possible partition

For a graph with n vertices, each possible partition of the set V of vertices into disjoint sets R, S and T, is represented by an n-digit ternary number. A bit set to 2 represents a vertex in the set R, a bit set to 1 represents a vertex in the set S, and a bit set to 0 represents a vertex in the set T. For example, the partition $R = \{A_4, A_5, A_6\}$, $S = \{A_2, A_3\}$, $T = \{A_1, A_7, A_8\}$ in Fig. 1 is represented by the ternary number 00222110. In this way, we transform all possible partition of V in an n-vertex graph into an ensemble of all n-digit ternary numbers. We call this the data pool.

(1-1) Merge (P, Q);

(1-2) Annealing (P);

(1-3) Denaturation (P);

(1-4) Separation ($P, \{A_1\#\}, T_{tmp}$);

(1-5) Discard (P);

(1-6) Separation ($T_{tmp}, \{\#B_n\}, P$).

After above six steps of manipulation, singled strands in tube P will encode all 3^n partitions of V in the form of n-digit ternary numbers. For example, for the graph in Fig. 1 with n8 we have, e.g. the singled strand
$\#B_8 0A_8 B_7 0A_7 B_6 2A_6 B_5 2A_5 B_4 2A_4 B_3 1A_3 B_2 1A_2 B_1 0A_1 \#$,
which denotes the partition $R = \{A_4, A_5, A_6\}$, $S = \{A_2, A_3\}$, $T = \{A_1, A_7, A_8\}$ corresponding to the ternary number 00222110.

Existence of $B_i k A_i$ in a singled strand, where k is equal to 0, 1 or 2 means vertex i is in T, S or R respectively.

This operation can be finished in $O(1)$ steps since each manipulation above works in $O(1)$ steps.

3.2 Select invalid partition

For each element in the data pool that represents some partition $\{R, S, T\}$ of V, we count number of edges that have one endpoint in R and have no one in S and vice versa. Let the partition $\{R, S, T\}$ correspond the n-digit ternary number $a_n \ldots a_i \ldots a_j \ldots a_1$. For each pair (a_i, a_j) with $a_i = 2, a_j = 1$ or $a_i = 1, a_j = 2$ we append the singled strand y_{ij} or y_{ji} to the end of the singled strand which encode the n-digit ternary number $a_n \ldots a_i \ldots a_j \ldots a_1$. For example, the singled strands $\#B_8 1A_8 B_7 1A_7 B_6 0A_6 B_5 1A_5 B_4 0A_4 B_3 2A_3 B_2 2A_2 B_1 2A_1 \#$ (representing the ternary number 11010222 for the graph in Fig. 1) is transformed into $\#B_8 1A_8 B_7 1A_7 B_6 0A_6 B_5 1A_5 B_4 0A_4 B_3 2A_3 B_2 2A_2 B_1 2A_1 \# y_{8,2} y_{8,1} y_{7,1}$ where the singled strands $y_{8,3}, y_{7,3}, y_{7,2}, y_{5,3}, y_{5,2}, y_{5,1}$ do not appear since there are not corresponding edges in the graph shown in Fig. 1 and so they all have length 0-mer by the definition of $y_{i,j}$. It means that if we take the partition $\{R, S, T\}$ of vertices of the graph in Fig. 1 to be $R = \{A_4, A_5, A_6\}$, $S = \{A_2, A_3\}$, $T = \{A_1, A_7, A_8\}$, then there are totally three edges $A_8A_3, A_7A_3, A_7A_2, A_5A_3, A_5A_2, A_5A_1$ which have one vertex in R and one in S. Every partition $\{R, S, T\}$ with an edge, has one endpoint in R and one in S is an invalid partition.

For k = 1 to k = n

(2-1) Separation ($P, \{B_k 1A_k\}, T_1$)

For i = 1 to i = n

(2-2) Separation ($T_1, \{B_i 2A_i\}, T_2$)

(2-3) Append ($T_2, y_{k,i}$)

(2-4) Merge (T_1, T_2)

End For

(2-5) Merge (P, T_1)

End For

(2-6) Separation ($P, \{y_{11}\}, T_1$)

(2-7) Discard (T_1)

In the above operation we use two "For" clauses. Thus this operation can be finished in $O(n^2)$ steps since each single manipulation above works in $O(1)$ steps.

3.3 Counting number of vertices of each subset and select invalid partition

For each element in the data pool that represents some partition $\{R,S,T\}$ of V, we count the vertices number of R, S and T. Let the partition $\{R,S,T\}$ correspond the n-digit ternary number $a_n \ldots a_i \ldots a_j \ldots a_1$. For each element a_i with $a_i = 2$ we append the singled strand 5 to the end of the singled strand which encode the n-digit ternary number $a_n \ldots a_i \ldots a_j \ldots a_1$ .and also For each element a_i with $a_i = 1$ we append the singled strand 4 and For each element a_i with $a_i = 0$ we append the singled strand 3 to the end of the singled strand which encode the n-digit ternary number $a_n \ldots a_i \ldots a_j \ldots a_1$. For example, the singled strands $\#B_8 0A_8 B_7 2A_7 B_6 2A_6 B_5 2A_5 B_4 2A_4 B_3 1A_3 B_2 1A_2 B_1 0A_1 \#$ (representing the ternary number 02222110 for the graph in Fig. 1) is transformed into $\#B_8 0A_8 B_7 2A_7 B_6 2A_6 B_5 2A_5 B_4 2A_4 B_3 1A_3 B_2 1A_2 B_1 0A_1 \#$ 55554433 .Every partition that $|R| \geq \beta |V|$ or $|S| \geq \beta |V|$ is invalid. Let $L = \lceil \beta |V| \rceil + 1$

For i = 1 to i = n

(3 - 1) Separation $(P, \{B_i 2A_i\}, T_1)$

(3 - 2) Append $(T_1, 5)$

(3 - 3) Merge (P, T_1)

End For

For i = 1 to i = n

(3 - 4) Separation $(P, \{B_i 1A_i\}, T_1)$

(3 - 5) Append $(T_1, 4)$

(3 - 6) Merge (P, T_1)

End For

For i = 1 to i = n

(3 - 7) Separation $(P, \{B_i 0A_i\}, T_1)$

(3 - 8) Append $(T_1, 3)$

(3 - 9) Merge (P, T_1)

End For

(3 - 10) Separation $(P, \{\underbrace{55...55}_{\text{L times}}\}, T_1)$

(3 - 11) Separation $(P, \{\underbrace{44...44}_{\text{L times}}\}, T_2)$

3.4 Finding the minimum size of subset T

Now, among all (valid) partitions, we separate the partitions in which the size of T is minimum.

For i = 1 to i = n

(4 - 1) Separation $(P, \{\underbrace{33...33}_{\text{i times}}\}, T_1)$

(4 - 2) Separation $(T_1, \{\underbrace{33...33}_{\text{i+1 times}}\}, T_2)$

If Detect (T_2) is "yes", then End For and i is the number of vertex corresponding to minimum β - vertex separator , else continue the circulation.

(4 - 3) Merge (P, T_2)

End For

3.5 Giving the exact solutions

Finally the *Read* operation is applied to giving the exact solutions to the minimum β -vertex separator problems.

(5 – 1)Read (T_2).

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 β -vertex separator NP-complete problems in the Adleman–Lipton model. The procedure works in $O(n^2)$ steps for minimum β -vertex separator problems of an undirected 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, 1021– 1024, 1994.

[2] Frisco, P., Parallel arithmetic with splicing. Romanian J. Inf.Sci. Technol. 2, 113–128, 2002.

[3] Fujiwara, A., Matsumoto, K., Chen, Wei. Procedures for logic and arithmetic operations with DNA molecules. Int. J. Found. Comput. Sci. 15, 461–474, 2004.

[4] Guarnieri, F., Fliss, M., Bancroft, C. Making DNA add. Science 273, 220–223, 1996..

[5] 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. BioSystem 80, 71–82, 2005.

[6] Gupta, V., Parthasarathy, S., Zaki, M.J. Arithmetic and logic operations with DNA. In: Proceedings of Third DIMACSWorkshop on DNA-Based Computers, 212–220, 1997.

[7] Hug, H., Schuler, R. DNA-based parallel computation of simple arithmetic. In: Proceedings of the Seventh International Meeting on DNA-based Computers, 159–166, 2001.

[8] Kamio, S., Takehara, A., Fujiwara, A. Procedures for computing the maximum with DNA strands. In: Arabnia, Humid, R., Mun, Youngsong (Eds.), In: Proceedings of the International Conference on DNA-Based Computers. 2003.

[9] Li, D., Huang, H., Li, X., Li, X. Hairpin formation in DNA computation presents limits for large NP-complete problems. BioSystem 72, 203–207, 2003.

[10] Lipton, R.J. DNA solution of HARD computational problems. Science 268, 542–545, 1995.

[11] Ouyang, Q., Kaplan, Peter, D., Liu, S., Libchaber, A. DNA solution of the maximal clique problem. Science 278, 446–449, 1997.

[12] Pâun, G., Rozeberg, G., Salomaa, A. DNA Computing. Springer-Verlag. 1998.