UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 7, July 2017

Fast Parallel Molecular Solution for Longest and Shortest Paths with Forbidden Pairs

Mahsa Hakimi[1], Akbar Moazzam[2], Babak Dalvand[3], Mehdi S Haghighat[4], Ahmed A. Sallam[5], Ardashir Dolati[6]

Department of Pharmacy, Shahid Beheshti University , Tehran, Iran[1]

Department of Computer Science, Isfahan University, Isfahan, Iran [2]

Department of Mathematics, Arak University, Arak, Iran[3,4]

Department of Computer Science, Cairo University, Giza, Egypt[5]

Department of Mathematics, Shahed University, Tehran, Iran[6]

Abstract: Adleman demonstrate that we can deoxyribonucleic acid (DNA) strands to solve an instance of the Hamiltonian path problem (HPP). One year later, Lipton solved another NP hard problem. In this paper, we use this model for developing a new DNA algorithm to solve longest and shortest path with forbidden pairs. Our algorithm works in polynomial time.

Keywords: DNA computing, longest and shortest path with forbidden pairs.

I. INTRODUCTION

With Watson-Crick complementarily and massive parallelism features of DNA, we can solve NP hard problems in linear or polynomial time. It is clear we do not have any polynomial algorithms to solve NP hard problems with silicon base computers; But DNA computing provides powerful features which can solve those problems in polynomial steps. Adleman [1] solved Hamiltonian path problem of size n. That was the first algorithm for DNA computing. Lipton [5] solved the second NP hard problem with those operations. Some other NP-hard problems which have been solved [6-21].

In this paper, the DNA operations proposed by Adleman [1] and Lipton [5] are used to solve Longest path with forbidden pairs.

For a, given Graph $G=(V,E)$ and a collection $C=\{(a_1,b_1),\ldots,(a_m,b_m)\}$ pairs of vertices from V. a minimum solution is simple path in G that contains at most one vertex from each pair in C with lonest path.

In Section 2, the Adleman–Lipton model is introduced in detail. Section 3 we will present a DNA algorithm for solving the Longest and shortest path with forbidden pairs problem and the complexity of the proposed algorithm is described. We give conclusions in Section 4.

II. 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 [14].

A DNA (deoxyribonucleic acid) is a polymer, which is strung together from monomers called deoxyribonucleotides [14]. Distinct nucleotides are detected only with their bases [13].

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 [12].

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 [4]. Thus, if we make a double strand from two single strands of length 20 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]. The DNA operations proposed by Adleman and Lipton [2] 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 [2]:

DOI10.17148/IARJSET.2017.4722

(1) Merge (T1, T2): for two given test tubes T1, T2 it stores the union $T_1 \cup T_2$ in T1 and leaves T2 empty [4];
(2) Copy (T1, T2): for a given test tube T1 it produces a test tube T2 with the same contents as T1 [2];
(3) Detect (T): Given a test tube T it outputs ‘‘yes’’ if T contains at least one strand, otherwise, outputs ‘‘no’’ [2];
(4) Separation (T1, X, T2): for a given test tube T1 and a given set of strings X it removes all single strands containing a string in X from T1, and produces a test tube T2 with the removed strands [3];
(5) Selection (T1, L, T2): for a, given test tube T1 and a given integer L it removes all strands with length L from T1, and produces a test tube T2 with the removed strands [8];
(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 $\left[\frac{\sigma_0\sigma_1}{\sigma_0\sigma_1}\right]$ in T into two double strands as follows:

$$\left[\frac{\alpha_0\sigma_0\sigma_1\beta_0}{\alpha_1\sigma_0\sigma_1\beta_1}\right] \Rightarrow \left[\frac{\alpha_0\sigma_0}{\alpha_1\sigma_0}\right], \left[\frac{\sigma_1\beta_0}{\sigma_1\beta_1}\right]$$

(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 [6];
(8) Denaturation (T): for a, given test tube T it dissociates each double strand in T into two single strands [7];
(9) Discard (T): for a, given test tube T it discards the tube T [11];
(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 [12];
Since these eleven manipulations are implemented with a constant number of biological steps for DNA strands, we assume that the complexity of each manipulation is $O(1)$ steps [14].

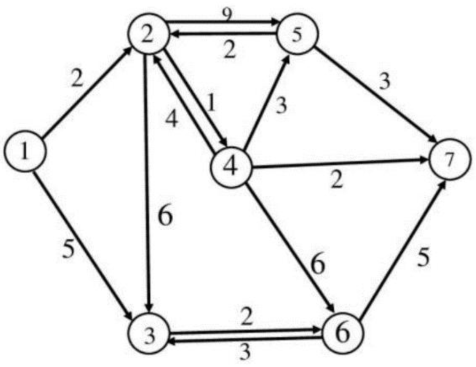

Fig. 1 Graph G.

III. SOLVING LONGEST AND SHORTEST FORBIDDEN PATH BY ADLEMAN-LIPTON MODEL

Let $G=(V,E)$ be a directed graph with the set of vertices being $V=\{A_k \mid k=1,2,\ldots,m\}$ and the set of edges being $E=\{e_i \mid i=1,2,\ldots,n\}$ [3]. Let |E|=d. In the following, the symbols $\#, X, Y, A_k, B_j\ (k=1,2,\ldots,m, j=1,2,\ldots,m)$ denote distinct DNA singled strands with same length, say 10-mer. And ||.|| 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 to distinguish all above symbols [3]. We assume the DNA singled strand $Y_{i,j}$ is used to denote the weights on the edges $e_{i,j}$ in E, so $||Y_{i,j}|| = w_{i,j}$.

Let $L = Max_{e_{i,j}\in E}\{w_{i,j}\}$ .Suppose that all weights in the given graph are commensurable, i.e., there exists a number y such that each weight is an integral multiple of y (here, take y=10) in the following discussion. In our previous work [11] we defined P and Q tubes and introduced an algorithm to produce all paths from a vertex to another one. We use that algorithm to produce all paths.

IV. PRODUCE ALL PATHS WHICH CONTAINS AT MOST ONE VERTEX IN EACH PAIR IN C

We introduced an algorithm [11] to encode all paths. for instance
$\#A_1B_1Y_{1,2}A_2B_2Y_{2,4}A_4B_4Y_{4,2}A_2B_2Y_{2,5}A_5B_5Y_{5,7}A_7B_7\#$ denotes 1→2→4→6→7.

In this step, we want to select all paths which have at most one vertex in each pair of C and remove the remaining set. If we have a pair like (5,2) in C, then 1→2→4→6→7 is acceptable answer. but for (4,1) is not acceptable because vertex 1 and 4 are in that path.
In Collection C, we show the i-th pair with Ci1,Ci2 which denotes the first and second element of i-th pair.

```
For $i = 1$ to $m$
    (1-1) Separation $(P, \{A_{ci1}B_{ci1}\}, T_1)$
    (1-2) Separation $(T_1, \{A_{ci2}B_{ci2}\}, T_2)$
    (1-3) Merge $(P, T_1)$
    (1-4) Discard $(T_2)$
End for
```

In 1-2, we choose all paths which contain the first element of i-th pair, and put them in T1, if any of those paths contains the second element of i-th path, it is invalid path and we need to remove it. Here we have simple loop then our algorithm is O(m) which is depends on the size of set C.

V. FIND THE LONGEST AND SHORTEST ANSWER

Each strand contains n number of A_iB_i and two # and the length of A_iB_i are 20. also, the length ||#|| is 10. Each strand has n number of $||Y_{i,j}|| = w_{i,j}$ then the maximum length of each strand is 20*n+20+n*L.
For instance, for this strand the maximum length is 20*7+20+7*L
$\#A_1B_1Y_{1,2}A_2B_2Y_{2,4}A_4B_4Y_{4,2}A_2B_2Y_{2,5}A_5B_5Y_{5,7}A_7B_7\#$

```
For $i = 1$ to $n*L/10$
    (2-1) Selection $(P, 20*(n+1)+n*L-10*i, T_1)$
    if Detect$(T_1)$
        exit()
End for
```

For finding the shortest one we use this algorithm.

```
For $i = 1$ to $n$
    (2-1) Selection $(P, 20*(n+1)+i*L, T_1)$
    if Detect$(T_1)$
        exit()
End for
```

L is a constant, then this algorithm will terminate in O(n). with this algorithm we can find shortest path and longest path in O(n).
Then our algorithm will terminate in O(m+n).

VI. CONCLUSION

In this paper, we proposed new polynomial algorithm for two of NP-Hard Problems.
As you can see this algorithm will finish in $O(n+m)$

REFERENCES

[1] Leonard M Adleman. Molecular computation of solutions to combinatorial problems. Nature, 369:40, 1994.
[2] Babak Dalvand, Saeed Safaei, and Mojtaba Nazari. Fast parallel molecular solution to the maximum triangle packing problem on massively parallel bio-computing. In FCS, pages 169–173, 2009.
[3] Ardashir Dolati, Mehdi Sohrabi Haghighat, Saeed Safaei, and Hajar Mozaffar. Solving minimum beta-vertex separator problems in the adleman-lipton model. In FCS, pages 97–101, 2008.
[4] Ardeshir Dolati, Mehdi S Haghighat, and Saeed Safaee. Solving bin-packing problem in the adleman–lipton model. In the First Conference and Workshop on Mathematical Chemistry, volume 1, pages 70–74, 2008.
[5] Richard J Lipton. Dna solution of hard computational problems. Science, 268(5210):542, 1995.
[6] Assefi, Mehdi, et al. "An Experimental Evaluation of Apple Siri and Google Speech Recognition." Proccedings of the 2015 ISCA SEDE (2015).

[7] Saeed Safaei, Babak Dalvand, Babak Esmaeili, and Vahid Safaei. Molecular solutions for the minimum edge dominating set problem on dna-based supercomputing. In FCS, pages 32–36, 2009.

[8] Saeed Safaei, Babak Dalvand, Nozar Safaei, and Vahid Safaei. Molecular solutions for the maximum k-facility dispersion problem on dna-based supercomputing. In FCS, pages 513–517, 2011.

[9] Saeed Safaei, Hajar Mozaffar, and Babak Esmaeili. Solving minimum k-center problem in the adleman? lipton model. In FCS, pages 251–255, 2008.

[10] Saeid Safaei, Babak Dalvand, Zahra Derakhshandeh, and Vahid Safaei. Molecular solutions for the bin-packing and minimum makespan scheduling problems on dnabased supercomputing. In PDPTA, pages 512–516, 2009.

[11] Sohrabi-Haghighat, M., Dolati, A., & Safaee, S. (2015). Solving several kinds of constrained shortest path problems via DNA. Applied mathematics in Engineering, Management and Technology, 527-533.

[12] Li, L., Wenhua, L., Wang, Z., Tunhua, W., & Ping, W. (2015). Solving the Maximum Weight Vertex Independent Problem with DNA Molecules in Adleman-Lipton Model. Journal of Computational and Theoretical Nanoscience, 12(8), 1940-1943.

[13] Wang, Z., Ji, Z., Su, Z., Wang, X., & Zhao, K. (2016). Solving the maximal matching problem with DNA molecules in Adleman– Lipton model.International Journal of Biomathematics, 9(02), 1650019.

[14] Zhao, K., Wang, Z., Lu, Y., Qin, J., & Tan, J. (2015). A new biologically DNA computational algorithm to solve the k-vertex cover problem. Journal of Computational and Theoretical Nanoscience, 12(3), 524-526.

[15] Wang, Z., Huang, D., Tan, J., Liu, T., Zhao, K., & Li, L. (2015). A parallel algorithm for solving the n-queens problem based on inspired computational model. BioSystems, 131, 22-29.

[16] Li, L., Zhao, K., & Ji, Z. (2015). A Genetic Algorithm to Solve the Subset Sum Problem based on Parallel Computing. Appl. Math, 9(2), 921-925.

[17] Zheng, X., Wang, B., Zhou, C., Wei, X., & Zhang, Q. (2016). Parallel DNA Arithmetic Operation With One Error Detection Based on 3-Moduli Set. IEEE transactions on nanobioscience, 15(5), 499.

[18] Assefi, M., Liu, G., Wittit, M. P., & Izurieta, C. Measuring the Impact of Network Performance on Cloud-Based Speech Recognition Applications.

[19] Assefi, M., Wittie, M., & Knight, A. (2015, August). Impact of Network Performance on Cloud Speech Recognition. In 2015 24th International Conference on Computer Communication and Networks (ICCCN) (pp. 1-6). IEEE.

[20] Moazzam, Akbar, and Babak Dalvand. "Molecular solutions for the Maximum K-colourable Sub graph Problem in Adleman-Lipton model." arXiv preprint arXiv:1610.07294 (2016).

[21] Sallam, Ahmed A., Mohamad Kazem, and Abdolah B. Askar. "Solving Minimum Independent Dominating Set with Adelman-Lipton model."

[22] Sallam, Ahmed A., and Salwa Badr. "Solving Minimum Cut Cover with Adelman-Lipton model."

[23] Fazli, Mojtaba Sedigh, and Jean-Fabrice Lebraty. "A comparative study on forecasting polyester chips prices for 15 days, using different hybrid intelligent systems." Neural Networks (IJCNN), The 2013 International Joint Conference on. IEEE, 2013.

[24] Fazli, Mojtaba Sedigh, and Jean-Fabrice Lebraty. "A solution for forecasting pet chips prices for both short-term and long-term price forecasting, using genetic programming." The 2013 International Conference on Artificial Intelligence. Vol. 2. CSREA Press, 2013.

[25] Liu, Yi, and Kwangjo Kim. "Understanding Web 2.0 service models, applications, reflections, perspectives, and beyond." PeerJ Preprints 5 (2017): e2811v1.

[26] Madani, Mirmorsal. "Comparing learning algorithms in neural network for diagnosing cardiovascular disease." arXiv preprint arXiv:1611.01678 (2016).

[27] Lee, Cheng. A survey of the World Wide Web evolution with respect to security issues. No. e2793v1. PeerJ Preprints, 2017.

[28] Soleimanian, Azam, and Ghasemali Salmani Jajaei. "Robust nonlinear optimization with conic representable uncertainty set." European Journal of Operational Research 228, no. 2 (2013): 337-344.