An Approach Towards the Solution of NP-Complete Problem

Kamran Khan Babar
Islamia University
Bahawalpur, Pakistan
Kamran.khan_2186@nadra.gov.pk

Abstract: *DNA Computing is hot topic and a substitute way for computational calculations. It is based on the statement that in general it is possible to design a sequence of biochemical tests including DNA elements which is comparative to processing information secured in these elements. Cook's Theorem informs that if one algorithm for an NP-complete or an NP-hard issue will be designed, then other issues will be fixed through reduction to that problem. The minimum vertex cover problem is a traditional classical graph optimization problem and has been proven to be NP-Complete problem. In this article, we present a DNA based algorithm for fixing the minimum vertex-cover problem.*

Keywords: DNA Computing, NP-complete Problems, NP-hard Problems, Minimum Vertex Cover Problem, Cook's Theorem

Received: 16 March 2014, Revised 29 April 2014, Accepted 4 May 2014

1. Introduction

DNA computing is a hot topic and an analysis research area that is growing fast now days. This is due the fact that the DNA molecules are implemented in a computational process. The main purpose of this research area is to produce, a biologically inspired computer which is totally based on DNA molecules in near future .This computer will replace a silicon based computer in near future. It took 30 years till Adleman in 1994 making proof of the principle study that DNA molecules can solve an NP problem of Hamiltonian Path through bio-chemical procedure [2] since after the discovery of R. Feynman to construct a computer from molecules in 1964 [1]. DNA is a basic storage medium for all of the living cells. The primary purpose of DNA is to absorb and transmit the data of life for trillions of the life years. Roughly, it is around 10 trillions of DNA molecules could fit into a space, the small size of a marble. Since all these DNA molecules can process data simultaneously, so we can calculate 10 trillions times of data simultaneously in a small space at one time. DNA computing is more generally known as molecular computing. It is interdisciplinary field where it is a combination of biology, chemistry, and mathematics and computer science. DNA Computing offers a completely new paradigm for computation. The basic purpose of DNA computing is to encode data in a DNA strand form. After this applying the laboratory techniques of molecular biology, called as bio operations, we can manipulate DNA strands in a test tube in order to simulate arithmetical and logical operations. It is estimated that a mix of 1018 DNA strands could operate 104 times faster than the speed of a today's advanced super computer [3].

Vertex cover is an NP-complete problem. We cannot expect to find a polynomial time algorithm for finding a minimum size vertex cover. The size of a vertex cover produced by the approximation algorithm is almost twice the minimum size of a vertex cover. The basic purpose of the vertex cove problem is to find a vertex cover of minimum size in a given undirected graph. We will call such a vertex cover an optimal vertex cover. This problem is the Optimization version of an NP-complete decision problem.We buy in

the argue that the common belief that NP-hard optimization problems cannot be solved exactly in polynomial time. So in the past a great research has been devoted in, to drive efficient approximation algorithm. i.e. algorithms that deliver solutions whose value is guarantee to be within some multiplicative factor from the optimum. In order to evaluate the able in polynomial time.

In the general case, a very simple 2-approximate algorithm has been known for thirty years [4], and no other better approximation algorithm has been found until now. Slightly better approximation guarantees are achievable over bounded degree graphs [5]. On the negative side, the minimum vertex cover problem has been shown to be Max SNP-hard even when restricted to graphs with maximum degree 3 by Papadimitriou and Yannaakasis [6]. Their reduction is from MAX 3-SAT and uses explicit construction of expander graphs [7]. Combining this reduction, the non-approximability results by Ballare et al. [8] and the performance guarantees of such approximation algorithms, it is important to understand how far we can go. i.e. to prove, for any approximable problem, which is the best approximation achieve.best known explicit construction of expanders [9], one can show that Minimum vertex cover is not 1.00036-approximableon bounded degree graphs. Ballare et al. [8] give a 1.0688 lower bound for the general minimumvertex cover problem by using a different technique, namely, they reduce directly from the computation of averifier using a somehow "*complementary*" version of the FLGSS reduction [10]. However, their method does not apply when classes of graphs in which a fixed bound on the maximum degree or some other density constraints are considered.

Minimum Vertex Cover Problem:

The minimum vertex cover problem arises in various important applications, including in multiple sequence alignments in computational biochemistry. Several approaches, such as the use of a parameterized algorithm [11] and the use of a simulated annealing algorithm [12], has also been developed to solve thisproblem. As the DNA computing, uses the concept of parallel computing, so it can be used to solve large problems. Now, this study introduces an alternative molecular computing approach to solve the minimum vertex cover problem.

A vertex cover for a graph G is a set of vertices S so that every edge of G is incident to at least one vertex in S. Namely, S covers the edges of G. The Minimum Vertex Cover problem is to find the minimum set of vertices that cover all edges. Given an undirected graph $G = (S, T)$, $m = |S|$ and $n = |T|$ are defined as the numbers of vertices and edges, respectively. A vertex edge incidence matrix $A = (a_{ij})$ of G is defined as $a_{ij} = 1$ if edge j is incident to vertex i; otherwise $a_{ij} = 0$, with $i = 1, \ldots, m$; $j = 1, \ldots, n$. The Minimum Vertex Cover problem can be stated as follows.

Vertex Cover Problem is NP-Complete

A vertex cover of an undirected graph $G = (S, T)$ is a subset $S' \in S$ such that if $(u, v) \in T$, then $u \in S'$ or $v \in S'$ (or both). That is, each vertex "*covers*" its incident edges, and a vertex cover for G is a set of vertices that covers all the edges in T. The size of a vertex cover is the number of vertices in it. The vertex-cover problem is defined as to find a vertex cover of minimum size in a given graph. Restating this optimization problem as a decision problem, we wish to determine whether a graph has a vertex cover of agiven size k. Since VERTEX-COVER is NP-complete, we don't expect to find a polynomial-time algorithm for finding a minimum-size vertex cover.

DNA algorithm for Vertex Cover Problem

The following DNA algorithm is proposed to solve the vertex-cover problem:

Step 1: Encoding of the problem in DNA

Encoding the vertices

For each vertex, synthesize a random 10-based palindrome DNA strand where S_i represents the i^{th} vertex.

Encoding the edges

For each directed edge S_i S_j, synthesize a 10-base DNA strand consisting complementary of 3' 5-mer sequence of S_i and complementary of 5' 5-mer sequence of S_j. Each vertex S_i in the graph has to be associated with a designed palindrome 10-mer sequence of DNA denoted by S_i. For each edge S_i, S_j *in* the graph, an oligonucliotide 3' 5-mer complementary sequence of S_i followed by 5' 5-mercomplimentary sequence of S_j to be synthesized.

Step 2: Create an empty set for vertices and take acopy of edge set

After the completing of step 1 (i.e. encoding the vertices and edges) we create an empty set for vertices (S) and edges (T) after

that we will create acopy of edge set (T').

Step 3: Repeatedly picks an edge (S_i, S_j) from the copy of edge set

To avoid repetition of the nodes in the DNA strands an effective method, single stranded conformation polymorphism SSCP technique has to be used. The mobility in gel electrophoresis of double stranded DNA's of a given length is relatively independent of nucleotide sequence. In contrast, the mobility of single strands can vary time to time as a result of only small changes in nucleotide sequence. This fact led to the development of single-stranded conformation polymorphism (SSCP) techniques [13]. SSCP is the simplest and mostly used method of mutation detection. PCR is used to amplify the region of interest and the resultant DNA can be separated as single-stranded molecules by electrophoresis in a no denaturing polyacrylamide gel. A strand of single-stranded DNA folds differently from another if it differs by a single base, and it is believed that changes of structure of the DNA results in radiograms (radioactive detection), by silver staining of bands or the use of fluorescent PCR primers which are subsequently detected by an automated DNA sequencer (non-radioactive detection). Since all the nodes encoded are palindrome DNA strands, repetition of the nodes can lead to formation of Hairpin loop structures [14]. These hairpins like structures have to be eliminated from single stranded DNA strands by the above process. The hairpin loop strands are not considered for deriving the solution.

Step 4: Amplification of DNA paths by PCR

Amplification of DNA paths has to be performed that begin with vertex source and end with vertex destination. Two specific primers that can anneal with source vertex and destination vertex are to be added to the PCR reaction.

Step 5: Mark the endpoints S_i and S_j to empty set

In step 4 we will pick an edge (S_i, S_j) from the copy of edge set (T'). Now we mark its end points say S_i and S_j to the set of vertices which was created in step 2as an empty set.

Step 6: Delete the edges from the copy of edge set

In this step we will remove all the edges from the copy of edge set (T') that are covered by either vertex S_i or vertex S_j. And this process will be repeated until the copy of edge set will be empty.

Step 7: Sequencing of DNA strands

The strands obtained in the step 5 are now need to be sequenced. The weights of the strands can be determined by reading the sequence. The set having the minimum number of vertices that covers entire graph is now our desired solution.

Concluding Remarks

This paper has proposed a faster approach for searching and finding the solution for minimum vertex cover problem by using the DNA Computing. Because of ability of high degree of parallelism of the DNA computing, it can overcome the difficulties which may be, can cause the problem due to silicon computers. However the extensive use of DNA computing principles for solving simple problems may not be suggestible due to its high cost. In order to make the DNA computing applicable in practice and cheap further research in the fields of Computer science and biology is very necessary. Computer science needs to develop more elaborated DNA algorithms, while better enzymes and protocols are also necessary in biology to manipulate DNA molecules more selectively with minimum rate of errors.

References

[1] Feynman, R. P. (1961). Miniaturization, New York, Reinhold, p. 282-296.

[2] Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems, Sciences, 266 (5187) 1021-1024.

[3] Kari, L. (1997). From micro-soft to bio-soft: Computing with DNA, Biocomputing and Emergent Computation: Proc. of the BCEC97, World Scientific, Skovde, Sweden, p. 146-164.

[4] Gavril, F. (1974). Manuscript cited in [18].

[5] Monien, B., Speckenmeyer, E. (1983). Some further Approximation algorithms for the vertex cover problem. *In*: Proceedings of CAAP83, p. 341-349. LNCS 159, Springer Verlag.

[6] Papadimitriou, C. H., Yannakakis, M. M. (1991). Optimization, Approximation and Complexity classes. *Journal of Computer and System Sciences*, 43, p. 425-440. Preliminary version in Proc. of STOC'88.

[7] Gabber, O., Galil, J. (1981). Explicit construction of linear sized super concentrators. *Journal of Computer and System Sciences*, 22, p. 407-425.

[8] Bellare, M., Goldriech, O., Sudan, M. (1995). Free bits, PCP's and non-approximability-towards tight results (3rd version). Technical Report TR95-24 Electronic Colloquium on Computational Complexity. Priliminary version in Proc. of FOCS'95.

[9] Lubotzky, A., Philips, R., Sarnak, P. (1998). Ramanujan Graphs.Combinatorica, 8, p. 261-277.

[10] Feige, U., Goldwasser, S., Lovasz, L., Safra, S., Szegedy, M. (1991). Approximating clique is almost NP-complete. *In*: Proceeding of the 32nd IEEE Symposium on Foundations of Computer Science, p. 2-12.

[11] Downey, R. G., Fellows, M. R. (1995). Fixed parameter tractability and completeness II: completeness for W [1] Theory, Comp. Sci., 141, p.1-2.

[12] Xu, X., Ma, J. (2006). An efficient simulated annealing algorithm for the minimum vertex cover problem, Neurocomputing, 69, p. 913-916.

[13] Martyn Amos, Gheorghe Paun, Grzezorg, Rozenberg, ArtoSalomaa. (2000). Topics in the theory of DNA computing, Theoretical computer science, 287, p. 3-38.

[14] Sakamoto, K., Gouzu, H., Komiya, D., Kiga, S., Yokoyoma, T., Yokomori, Hagiya, M. (2000). Molecular computation by Hairpin formation, *Science*, 288, p. 1223-1226.

[15] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. Walter, P. (2002). *Molecular Biology of the Cell* (Garland, New York), 4th Ed.

[16] Storhoff, J. J., Mirkin, C. A. (1999). *Chem. Rev.* 99, p. 1849–1862.

[17] Bashir, R. (2001). *SuperlatticesMicrostruct.* 29, p. 1–16.

[18] Seeman, N. C. (1998) . *Angew. Chem. Int. Ed.* 37, p. 3220–3238.

[19] Winfree, E., Liu, F. R., Wenzler, L. A., Seeman, N. C. (1998). *Nature* 394, p. 539–544.

[20] Niemeyer, C. M., Burger, W., Peplies, J. (1998). *Angew. Chem. Int. Ed.* 37, p. 2265–2268.

[21] Cassel, A. M., Scrivens, W. A., Tour, J. M. (1998). *Angew. Chem. Int. Ed.* 37, p. 1528–1531.

[22] Mirkin, C. A., Letsinger, R. L., Mucic, R. C., Storhoff, J. J. (1996). *Nature* 382, p. 607–609.

[23] Braun, E., Eichen, Y., Sivan, U., Ben-Yoseph, G. (1998). *Nature* 391, p. 775–778.

[24] Mao, C., Sun, W., Shen, Z., Seeman, N. C. (1999). *Nature* 397, p. 144–146.

[25] Yurke, B., Turberfield, A. J., Mills, A. P., Jr., Simmel, F. C., Neumann, J. L. (2000). *Nature* 406, p. 605–608.

[26] Yan, H., Zhang, X., Shen, Z., Seeman, N. C. (2002). *Nature,* 415, p. 62–65.

[27] Lebedeva, I., Stein, C. A. (2001). *Annu. Rev. Pharmacol. Toxicol.*41, p. 403–419.

[28] Adelman, L. M. (1994). *Science*, 266, p. 1021–1024.

[29] Lipton, R. J. (1995). *Science*, 268, p. 542–545.

[30] Ouyang, Q., Kaplan, P. D., Liu, S., Libchaber, A. (1997). *Science* 278, p. 446–449.

[31] Khodor, J., Gifford, D. K. (1999). *Biosystems*, 52, 93–97.

[32] Ruben, A. J., Landweber, L. F. (2000). *Nat. Rev. Mol. Cell Biol.* 1, p. 69–72.

[33] Sakamoto, K., Gouzu, H., Komiya, K., Kiga, D., Yokoyama, S., Yokomori, T., Hagiya, M. (2000). *Science*, 288, p. 1223–1226.

[34] Faulhammer, D., Cukras, A. R., Lipton, R. J., Landweber, L. F. (2000). *Proc.Natl. Acad. Sci. USA* 97, p. 1385–1389.

[35] Mao, C., LaBean, T. H., Reif, J. H., Seeman, N. C. (2000). *Nature* 407, p. 493–496.

[36] Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Livneh, Z., Shapiro, E. (2001). *Nature* 414, p. 430–434.

[37] Braich, R. S., Chelyapov, N., Johnson, C., Rothemund, P. W. K., Adleman, L. (2002). *Science*, 296, p. 499–502.

[38] Ellington, A., Pollard, J. D., Jr. (1998). in *Current Protocols in MolecularBiology*, eds. Ausubel, F. M., Brent, R., Kingston. R. E., Moore, D. D., Seidman, J. G., Smith, J. A., Struhl, K. (Wiley, New York), p. 2.12.1–2.12.7.

[39] Chory, J., Pollard, J. D., Jr. (1998). in *Current Protocols in Molecular Biology*, eds. Ausubel, F. M., Brent, R., Kingston. R. E., Moore, D. D., Seidman, J. G., Smith, J. A. Struhl, K. (Wiley, New York), p. 2.7.1–2.7.8.

[40] Dickson, K. S., Burns, C. M., Richardson, J. P. (2000). *J. Biol. Chem.* 275, p. 15828–15831.

[41] Turing, A. M. (1936). *Proc. London Math. Soc*. 42, p. 230–265.

[42] Hopcroft, J. E., Motwani, R., Ullmann, J. D. (2000). *Introduction to Automata Theory, Languages, and Computation* (Addison–Wesley, Boston), 2nd Ed.

[43] Zamore, P. D. (2001). *Nat. Struct. Biol.* 8, p. 746–750.

[44] Kim, S. C., Skowron, P. W., Szybalski, W. (1996). *J. Mol. Biol.* 258, p. 638–649.

[45] Bennett, C. H. (1982). *Int. J. Theor. Phys.* 21, p. 905–940.

[46] Feynman, R. P. (1999). *In*: *Feynman Lectures on Computation*, eds. Allen, R. W., Hey, A. J. G. (Perseus, Cambridge, MA).

[47] Landauer, R. (1961). *IBM J. Res. Dev.* 3, p. 183–191.

[48] Keyes, R. W., Landauer, R. (1970) *IBM J. Res. Dev.* 14, p. 152–156.

[49] Bennett, C. H. (1973). *IBM J. Res. Dev.* 17, p. 525–532.

[50] Bennett, C. H. (1988). *IBM J. Res. Dev.* 32, p. 16–23.

[51] Fredkin, E., Toffoli, T. (1982). *Int. J. Theor. Phys.* 21, p. 219–253.

[52] Benioff, P. (1982). *Phys. Rev. Lett.* 48, p. 1581–1585.

[53] Zurek, W. H. (1989). *Nature* 341, p. 119–124.

[54] Li, M., Vita´nyi, P. M. B. (1996). *Proc. R. Soc. London Ser. A* 452, p. 769–789.