Molecular Computing Viability for Solving Computational Problems (Future and Challenges)

Hala M. AlShamlan and Mohammed El Bachir Menai

***Abstract*—Molecular computing is a field with a great potential, but few results of practical value. Although the area still remains to be one of the fastest growing fields of Computer Science it keeps the main issue of computation unsolved or in other word no polynomial time solution to NP-complete problems appeared. Although some approaches to solve NP complete problems were indeed successfully accomplished on DNA strands. In this paper we introduced the recent works accomplished using DNA computing for solving computational problem. After that, we investigate the open problems on DNA computing on the surface and the research directions in this area. Also, we present the main challenges and limitation of Molecular computing in specific "DNA computing ". Then, we give our analysis and potential proposal and suggestion about the viability of DNA computing to solve computational problems**

***Index Terms*—DNA computing, molecular computing, NP hard problem, and computational problems**

I. Introduction

Traditional Computing associated with solving a particular problem on the computer, nowadays is becoming one of the most applied branches of Computer Science. Modern multifunctional computers amaze by their capacity to perform calculations, and thus provide large spectrum of possibilities for solving computational problems coming from other sciences like biology or physics. However, Molecular computing has been developed for decades, and provides numerous ways for solving the difficult mathematic problems. All forms of molecular computing are currently in their infancy, but in the long run are likely to replace traditional silicon computers, which suffer barriers to higher levels of performance.

Molecular computing is a generic term for any computational scheme, which uses individual molecules as a means of solving computational problems. Molecular computing is most frequently associated with DNA computing, because that has made the most progress, but it can also refer to quantum computing or molecular logic gates. DNA computing, or more generally molecular computing, is based on manipulations with DNA strands using some basic biological transformations. Being very similar to parallel computing, DNA computing promises to solve many NP-complete problems, much faster than modern silicon-based computers do.

In DNA computing, DNA represent as the software whereas enzymes represent as the hardware. Custom-synthesized DNA strands are combined with enzymes in a test tube, and depending on the length of the resulting output strand, and then a solution can be derived. DNA computation is extremely powerful in its potential, but there are major drawbacks. DNA computation is non-universal, meaning that there are problems it cannot solved in principle. It can only return yes-or-no answers to computational problems. In 2002, researchers created a DNA computer, which could perform 330 trillion operations per second, more than 100,000 times faster than the speed of the fastest PC at the time.

Another proposal for molecular computing is quantum computing. Quantum computing takes advantage of quantum effects to perform computation, but the details are complicated. Quantum computing based on supercooled atoms locked in entangled states with one another. A major challenge is increasing of the number of computational elements (qubits) , it becomes progressively more difficult to insulate the quantum computer from matter on the outside, causing it to decohere, eliminating quantum effects and restoring the computer to a classical state. This ruins the calculation. Quantum computing may yet be developed into practical applications, but many physicists and computer scientists remain skeptical.

Generally, Molecular Computing problems are solved in three phases: encoding that maps the problem onto such as DNA strands, molecular operations that perform the basic core processing, and extraction/detection that makes the results visible to the naked eye [6]. However, molecular computing frequently used affinity purification has an accepted error rate of 5%, that means, it does not give much guarantee of the successful result of the computation. The amount of DNA molecules needed to solve Hamiltonian Graph Problem for 200 nodes exceeds the mass of our planet thus making it simply impractical [11] . In this paper we discuss the Molecular Computing in general and we pay more attention on the DNA computing because it is more popular than other molecular computing types, and there are a lot of applications based on DNA computing.

The rest of this paper organized as follow: in Section 1 we present general background about the initialization of molecular computing. After the background in Section 2 we give the reader literature review about DNA Computing. . Moreover, In Section 3, we investigate the open problems on DNA computing on the surface and the research directions in this area. In Section 4, we give the reader our analysis and expectation in this area also we present the main limitation and challenges. Finally in section 5, our conclusion and contribution is appears.

II. Background

First Molecular computing proposed in the 1970s, quantum computing relies on quantum physics by taking

Manuscript received May 1, 2012; revised June 2, 2012.

The authors are with Computer Science Department, King Saud University, Riyadh, Saudi Arabia (e-mail: halshamlan@ksu.edu.sa, menai@ksu.edu.sa)

advantage of certain quantum physics properties of atoms or nuclei that allow them to work together as quantum bits, or qubits, to be the computer's processor and memory. In addition, qubits can perform certain calculations exponentially faster than conventional computers by interacting with each other while being isolated from the external environment [30].Qubits do not based on the traditional binary nature of computing. A quantum computer can do an arbitrary reversible classical computation on all the numbers simultaneously, which a binary system cannot do. A quantum computer has some ability to produce interference between various different numbers By doing a computation on many different numbers at once, then interfering the results to get a single answer, a quantum computer has the potential to be much more powerful than a classical computer of the same size. In using only a single processing unit, a quantum computer can naturally perform myriad operations in parallel [30].Moreover, Quantum computing is not well suited for tasks such as word processing and email, but it is ideal for tasks such as cryptography and modeling and indexing very large databases.

In other hand, DNA computing is a form of computing which uses DNA and biochemistry and molecular biology, instead of the traditional silicon-based computer technologies. DNA computing, or, more generally, molecular computing, is a fast developing interdisciplinary area. Leonard Adleman of the University of Southern California initially developed this field. In 1994, Adleman demonstrated a proof-of-concept use of DNA as a form of computation which solved the seven-point Hamiltonian path problem. Particularly, since Adleman solved a small instance of the Hamilton path problem successfully, the DNA computing has become a new focus in the scientific areas of nanotechnology, biology, mathematics, medicine and information science [4] . In other hand, The Princeton group has provided evidence rely on detailed experimental that RNA may be more suitable to solve some computational problems because of its versatility. They argue that RNA is easier to use in substractive protocols that solve the problem by eliminating molecules representing problem constraint violations, rather than Adleman's additive method of building up the solution from basic building strands [5].

When DNA is used to compute solutions to hard NP complete problems, the output is read using a sequencing paradigm. However, silicon-based computation is achieved using binary logic, where the input and output are in a binary format. Therefore there was also interest to explore DNA's ability to compute within the same paradigm of binary logic, where the output is also in a binary format. As a testament to DNA's universal applicability to computation, this was achieved in two ways:

1) Construction of logic gates using DNA and
2) Logical self-assembly of DNA into specific super architecture

Logic gates using DNA: Small computational units, which control the processing of information according to a set of operations, are called logic gates. A logic gate senses one or more inputs and based on the processing information that they contain, produce an output. A biochemical reaction on DNA can also function as a logic gate, where the reactants are inputs, products are the output and the actual reaction is the operation. Here DNA assemblies contain fluorescent tags that are either activated or deactivated (the output) based on a trigger (the input). Logical assembly using DNA: The second approach to demonstrate DNA's capability of binary logic uses as its input, combinations of different DNA assemblies that can further self assemble according to the input logic into different kinds of super-architecture that may be visualized using high-resolution microscopy [6]. Using base-pair recognition of complementary sequences, it is possible to form discrete DNA assemblies of pre-defined shapes or motifs that are programmed with single stranded overhangs. [13]

Moreover, Protein bacteriorhodopsin is an older alternative to DNA molecules that support optical computing, In essence, this molecule contains the light sensitive rhodopsin present in vertebrate retinas. Also, it is consists of seven alpha-helical segments that span the purple membrane of a microorganism commonly known as *halobacterium halobium*. In addition, switching can take place by absorption of green and blue light as many as 10M times before wearing out. The switching property has been used in combination with lasers to create a storage medium for optical computer memories that is almost in the commercial stage now. The possibility exists that it might become a core memory for a molecular computer [5]. Although certainly involving amino acids at the protein-binding sites, this type of computation is more passive than the type described earlier. We thus use the expression "molecular computing" in order to avoid excluding any future developments with other media, but with the understanding that, currently, it essentially means DNA-based computing.

III. Literature Review

In 1994, Adleman [1] reported how to solve an instance of Hamiltonian path problem (HPP) by presenting a DNA-based polynomial-time method. HPP is to find an air flight path from given cities such that each city is visited once and only once. HPP is NP-complete. After that a major goal of subsequent research is how to use DNA manipulations to solve NP-hard problems, especially 3-SAT problems [17]. 3-SAT is the hardest of all NP problems, which is to search for a model (or solution) of a set of clauses with each clause composed of no more than three literals, where a literal is a variable (or an atom) or its negation. Various solutions were tried to solve the 3-SAT problem, Lipton [6] proposed DNA experiments on test tubes to solved a satisfiability problem based on DNA sticker computing model . Later, Ouyang used short linear dsDNA molecules and DNA restriction enzymes to solve maximal clique problem [7]. After that, another way for DNA computing was developed, In 2000, Liu et al [11]. introduced a new simple case and method to solve a 3-SAT problem, in which the feasibility of DNA surface computing was verified and also proved that the fluorescence could be used accurately in DNA computing. Furthermore, many previous DNA computing works still have lots of problems, such as the difficult controls of temperatures and specific recognition. Faced with those problems, there are also some improvements. Lee et al in [12]. proposed an encoding way, in which gradient temperatures were used . Thus, different DNA strands melting could be controlled by higher or lower temperatures. Recent years, with the development of DNA self-assembly, more multilevel and ordered DNA molecules have been utilized. The method of DNA self-assembly provides new instrument for DNA computing. Especially, DNA three-way branch migration is a novel method applied in DNA nanotechnology [14].

In 2002, researchers from the Weizmann Institute of Science in Rehovot, Israel, unveiled a programmable molecular computing machine composed of enzymes and DNA molecules instead of silicon microchips. The computer could perform 330 trillion operations per second, more than 100,000 times the speed of the fastest PC [27]. On April 28, 2004, Ehud Shapiro, Yaakov Benenson, Binyamin Gil, Uri Ben-Dor, and Rivka Adar at the Weizmann Institute announced in the journal Nature that they had constructed a DNA computer. This was coupled with an input and output module and is capable of diagnosing cancerous activity within a cell, and then releasing an anti-cancer drug upon diagnosis [28]. Donald Beaver in [2] design a molecular Turing machine based on interactions among DNA molecules. Unlike the non-universal methods described in [1], his methods specify a universal computing device capable of maintaining a state and memory and performing an indefinite number of transitions. Each computing device consists of single DNA molecular. This is means, many different molecules, encoding many different machines in arbitrarily different configurations, can be located in the same mixture and induced to undergo state transitions simultaneously Moreover, the chemical mechanisms for sate transitions permit parallel, heterogeneous, synchronized, computation.

In 2002, Karl-Heinz [10] applies DNA algorithms based on the sticker model to perform encoding, minimum-distance computation, and maximum-likelihood (ML) decoding of binary linear codes. In 2004, Shudong Wang [29] proposed solution based on the plasmid DNA computing models for the Minimum Vertiex Covering problem of graph. In 2009, the authors in [24] propose a method using plasmids to screen out the efficient solutions and delete the non-solution. The researcher in this paper, use one special separation device to pick up solution. First of all, encode the issues. Encode the vertexes and weights of edges by DNA fragments in directed graph, while the direction of edges can obtain through the vertex encoding. Then put these DNA fragments into solution to carry out biochemical reactions. Through basic biological operations such as connection reaction produce all random paths, they use plasmid to screen and then amplification, restriction, and finally pick up the shortest path. Also, in 2009, Qing-Hu Wang [25] proposed a computing method that based on incomplete Molecule Commixed Encoding (IMCE). In this method, the vertex of the graph is encoded as single-strand, weight is encoded as double-strands and the edge is incomplete encoded. This method is used to solve the shortest path problem. In 2009, Xiyu Liu [8] propose a migrating DNA computing model with built-in DNA computing engine to cluster analysis. This new technique will apply for large scale, high parallel clustering problems potentially

Regarding to Graph Colouring Problem, witch is one of the NP-complete problem ,Liu Xikuiin 2007 [21], provide DNA algorithms for Colouring Problems based on the molecular biology techniques to compute the vertex chromatic number of a given graph. The algorithms determine not merely the existence of a solution but yield all solutions (if any). The algorithm is highly parallel and has satisfactory fidelity. This research paper represents further evidence for the ability of DNA computing. Moreover, in 2009, Zhang Cheng [14] proposed a model to solve a 3-coloring graph problem, one of the NP problems, using the method of circular DNA displacement. The whole computing process is based on single circular DNA branch migrations just in a few tubes. For this computing model, two features are circular DNA and rather lower experimental complexity. Moreover, the key methods in this model include displacements of circular ssDNA (single strand DNA) and backtracking deletion algorithm. By repeating of DNA displacements, the correct solutions will be found after computing processes. For a 3-coloring graph problem with n vertices, using this algorithm, the time and space complexity both are O(n2) at most. During the computing course, an effective DNA strand displacement and exclusive sequence recognition ensure the accuracy of computing results. This model demonstrates circular DNA may have more applications in molecular computing.

IV. The Future of Research Direction in Molecular Computing

Molecular computing is a promising method for unconventional computation, owing to its merits of massive parallelism and efficiency in NP problem solving. One of the most challenging topics in the field of molecular computing is how to obtain an efficient degree of spatial complexity in ''manufacturing'' the molecules. Here the word ''manufacturing'' refers to the tasks for preparing or producing the materials by certain technical methods that will be used to build a molecular computer. Therefore, the major problem for Adleman's and Lipton's DNA computing experiments, is the time involved in extracting and recombining DNA. While DNA processes within the test-tube can take place millions of times per second, extraction processes, whereby individual strands of DNA are manually isolated and spliced, can take several hours and even days, just for the simplest problems. Thus, if we are to apply molecular computing algorithms to the processes of NP-complete problem solving, we really need to obtain a linear order in the space of controlling (i.e., the number of molecules to be controlled) under the condition of linear time complexity. This has led several researchers to conclude that the complexity aspects of DNA algorithms will limit their applicability.

One of the related open problems with DNA computing surface is develop standard format for DNA data representation, because there is no universal method of data representation. In today's computer systems, for example, the binary representation is universally agreed upon. DNA computing, however, has no such standard. Moreover, current research in DNA computing uses DNA as a data structure ('representational DNA'), as for instance above, where DNA is used to represent a map. But any algorithm that only assumes manual manipulation of data representations is unlikely to fare well in terms of time taken to produce a result. Instead, the issue is whether all the steps involved in algorithms for manipulating representational DNA can themselves be 'automated'.

Finally, one of the biggest problems facing the field of DNA computing is that no efficient implementation has been produced for testing, verification, and general experimentation. While Adleman's initial experiment was performed in a lab, many of the subsequent algorithms in DNA computing have never been implemented or tested. For this reason, in future research, we need real experimental DNA algorithm to give measurable and meaningful result.

V. Challenges and Limitation

One question must have emerged in the reader's mind after the discussion in the previous sections: how can all this potential of molecular computing be fully realized in real life? . For this reason in this section, we examine fundamental challenges that will need to be resolved for bringing molecular computing to an effective new paradigm for computational science.

A. Reliability, Efficiency, and Scalability

There are three most burning issues for molecular computing, which are reliability, efficiency, and scalability. The reliability of a protocol, i.e., a DNA computation, is the confidence degree with which a lab experiment provides a true answer to the given problem. The efficiency of the protocol measures the intended and effective use of the molecules that intervene in it. The scalability of a lab experiment is refer to the effective reproducibility of the experiment with longer molecules that can encode larger problem instances while still obtaining equally reliable results under comparable efficiency. These three are clearly interrelated problems. Because in Biological field the definition of success is different than in computer science, the biologists have not really faced these problems in their work [5]. Moreover, Research on these problems in molecular computing is in the beginning, and the most work has concentrated on reliability, and the encoding problem.

B. The Encoding Problem

When we chose the molecules for encoding the problem input, a molecular computer scientist is at the clemency of the chemistry, even though she may still have some control over the protocols that she may perform with them in the laboratory execution. However, the experiment can be repeated many times if the encodings are prone to errors, and always provide the same (erroneous) results, as evidenced in [5]. Therefore, this fact lessens the effectiveness of the standard method of increasing the reliability of a probabilistic computation with a nonzero probability of errors by iteration, and contemplated.

C. Building and Programming Molecular Computers

In a molecular computer, we can expect to find the basic features that are evident in a conventional electronic computer in an integrated system, namely information storage, programmability, and information processing. Some features are clearly desirable, but whether they are actually realizable is not very obvious.

Starting with the difficulties for implementing traditional algorithms in DNA and their potential for evolutionary-style computation, DNA computers obviously follow Michael Conrad's trade-off principle descried in [5] "a computing system cannot at the same time have high programmability, high computational efficiency, and high evolutionary adaptability." He defined programmability as the ability to communicate programs to the computing system exactly with a finite alphabet in a finite number of steps. The efficiency of a computing system is described as the ratio of the number of interactions in the system that are used for computing and the total number of interactions possible in the system, and the evolutionary.

D. Our Potential Proposal and Analysis

Despite all of the difficulties outlined above, there are still a number of researchers working on topics related to DNA computing. While they number fewer than in years past, much of their research seems to be motivated with a ground-up approach, focused on answering basic questions about DNA computing. Some more recent work has attempted to address the issues of data representation [5] and others with the ability to emulate today's circuit-based computing in a DNA-based system.

Generally, while a desktop PC is designed to perform one calculation very fast, DNA strands produce billions of potential answers simultaneously. This makes the DNA computer suitable for solving "fuzzy logic" problems that have many possible solutions rather than the either/or logic of binary computers. In the future, some speculate, there may be hybrid machines that use traditional silicon for normal processing tasks but have DNA co-processors that can take over specific tasks they would be more suitable for.

Finally, It remains to be seen whether or not DNA computing will become a viable method of problem solving in the future, but it should be clear that the momentum of quantum computing continues to grow at the expense of DNA-based methods. As outlined earlier, the advantage of massive parallelism that makes DNA computing seem so beneficial would also be provided by a quantum computer, should one be built. It seems unlikely that a case could be made for continuing research in DNA computing, given all its inefficiencies, if a reasonable implementation of quantum computing could be made.

VI. Conclusion

Currently, molecular computing is a field with a great potential, but few results of practical value. Although the area still remains to be one of the fastest growing fields of Computer Science it keeps the main issue of computation unsolved or in other word no polynomial time solution to NP-complete problems appeared. Although some approaches to solve NP complete problems were indeed successfully accomplished on DNA strands. DNA computing fundamentally being similar to parallel computing provides a nice way to make trillions of similar calculations in a moment, but nothing practically new in the eyes of computability theory. In this context it would be appropriate to say: DNA computer is fast, but not as fast as quantum.

Moreover, Digital computer provides a way to interact with its processor and memory in such a way that modern programmer simply writes lines of code in some high level language organizing loops, control flow statements and declaration of variables. While silicon-based computers take programmer away from basic operations, DNA computer does not have this ability, – to solve a particular problem on DNA molecules one should perform its simplest operations himself spending time in the laboratory.

Finally .Our contribution and conclusion for DNA computing abilities to solve NP-complete problem as follows:

DNA Advantage in solve NP –complete Problems

1) Perform million of operations simultaneously
2) Conduct large parallel processing
3) Massive amount of working memory
4) Generate and use own energy source via the input
5) Four storage bits A T G C
6) Minimization of data storage

DNA limitations in solve NP –complete Problems

1) DNA computing involve a relative large amount of error
2) Requires human assistance
3) Time consuming laboratory procedures
4) No universal method of data representation

Generally, it should be noted that now DNA computing is mostly applied on some types of problems while the universal problem solving tool is still to be created. The most skeptical conclusion one might make at the moment is that DNA computer is unlikely to substitute a regular one, rather to be confined to special problems. The more optimistic view would be the following: creating a functional DNA "computer" of the type most people are familiar with is possible, but lies many years in the future.

Acknowledgment

The authors would like thanks Research Center in King Saud University, Computer Science Department for continuous support and cooperation.

References

[1] L. M. Adleman, “Molecular Computation of Solutions to Combinatorial Problems,” Science, pp. 1021-1024 (Nov. 11, 1994).

[2] B. Donald, “Molecular Computing,” NSF, 1995.

[3] Q. Jian and S. Katsunori, “Signaling-pathway-based molecular computing for efficient 3-SAT problem solving,” *Information Sciences*, vol. 161, pp.121–137, 2004.

[4] S. Suruchi, B. Dhiraj, and K. Yamuna, “Transforming bases to bytes: Molecular computing with DNA,” *Current trend in science*, 2009.

[5] H. Max and J. Russell, “Biomolecular Computing and Programming,” *IEEE Transactions On Evolutionary Computation*, vol. 3, no. 3, 1999.

[6] R. J. Lipton, “DNA solution of Hard Combinatorial Problems,” Science, pp. 542-545, 1995.

[7] Q. Ouyang, P. D. Kaplan, and S. Liu, et. Al, “DNA Solution of the Maximal Clique Problem,” Science. pp. 446-449, 997.

[8] L. Xiyu and T. Hao, “A Migrating DNA Computing Technique for Grid Clustering,” *IEEE Transactions on evolutionary computation*, 2009.

[9] M. Dónail and D. A, “On the Scalability of Molecular Computational Solutions to NP Problems,” *J. Universal Computer Science. Springer*, vol. 2. no. 2, pp. 87-95, 1996.

[10] K. Z. Heinz, “On Applying Molecular Computation to Binary Linear Codes,” *IEEE Transactions On Information Theory*, vol. 48, no. 2, 2002.

[11] Q. Liu, et al, “DNA computing on surfaces,” *Nature*, pp. 175–179, 2000.

[12] J. Lee, S. Shin, I. Park, and I. Zhang, “Solving traveling salesman problems with DNA molecules encoding numerical values,” *Bio Systems Journal*, vol. 78, pp.39-47, 2004.

[13] J. M. Tour, W. L. V. Zandt, C. P. Husband, S. M. Husband, and L. S. Wilson, “Nanocell Logic Gates for Molecular Computing,” *IEEE Transactions On Nanotechnology*, vol. 1, no. 2, 2002.

[14] Z. C. Y. Jing and X. Jin, “A molecular computing model for 3-coloring graph problem based on circular DNA branch migration,” *IEEE Transactions on Evolutionary Computation*, 2009.

[15] L. Adleman, “Computing with DNA,” *Scientific American*, 1998.

[16] A. Martyn, “DNA Computing” Invited article for the Encyclopedia of Complexity and System Science,” Springer, 2008.

[17] M. Arita, N. Akio, and M. Hagiya, “Improving sequence design for DNA computig,” *Japan society for the Promotion of Science*, 2000.

[18] M. Arita, S. Akio, and M. Hagiya, “A Heuristic Approach for Hamilton Path Problem with Molecules,” *Japan society for the Promotion of Science*, 1997.

[19] J. Natasa and A. Stephen, “Graph Structure in DNA Computing,” University of south Florida press, 1998.

[20] R. Shalini, S. Vijay, and H. Naveen, “biological computer model to solve np complete problem,” *International Journal of Information Technology and Knowledge Management*, vol. 4, no. 1, pp. 191-194, 2011.

[21] L. Xikui and L. Yan, “Efficient DNA Algorithms for Chromatic Number of Graph Problems,” in *Proceedings of the IEEE International Conference on Automation and Logistics*, 2007.

[22] D. Li, “Is DNA computing viable for 3-SAT problems?” *Theoretical Computer Science, Elsevier*, vol. 290, pp. 2095 – 2107, 2003.

[23] Z. I, T. Yusei, and O. Osamu, “Direct-Proportional Length-Based DNA Computing for Shortest Path Problem,” *International Journal of Computer Science & Applications*, vol. I, no. 1, pp. 46 – 60, 2004.

[24] W. Zhenye, Q. Zhang, and Y. Dang, “The Improvement of DNA Algorithm to the Directed Shortest Hamilton Path Problem,” *IEEE Transactions on Evolutionary Computation*, 2009.

[25] W. Q. Hu, L. Zhi, P. Xiu, Q. Sun, and Z. Hong, “DNA algorithm based on incomplete molecule commixed encoding for the shortest path problem,” *The 1st International Conference on Information Science and Engineering*, 2009.

[26] N. Ajit and Z. Spiridon, “DNA algorithms for computing shortest paths” University of Exeter press, 1998.

[27] L. Stefan. "Computer Made from DNA and Enzymes," *National Geographic. Retrieved*, 2009.

[28] Y. Benenson1, B. Gil, U. B. Dor, R. Adar, and E. Shapiro, "An autonomous molecular computer for logical control of gene expression," *Nature journal*, vol. 429, pp.423–429, 2004.

[29] S. Wang, W. Liu, and J. Xu, "A DNA computing model minimal covering by plasmid," *Journal of Huazhong University of Science and Technology,* vol. 32, pp. 59-61, 2004.