DNA-Based Computing of Strategic Assignment Problems

Jian-Jun Shu,* Qi-Wen Wang, and Kian-Yan Yong
School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
(Received 9 March 2011; published 3 May 2011)

DNA-based computing is a novel technique to tackle computationally difficult problems, in which computing time grows exponentially corresponding to problematic size. A strategic assignment problem is a typical nondeterministic polynomial problem, which is often associated with strategy applications. In this Letter, a new approach dealing with strategic assignment problems is proposed based on manipulating DNA strands, which is believed to be better than the conventional silicon-based computing in solving the same problem.

DOI: 10.1103/PhysRevLett.106.188702 PACS numbers: 89.20.Ff

Introduction.—Since the first silicon-based microcomputer was first introduced at the beginning of the 1970s, improving computational ability has always been the top-ranked concern among researchers. However, no matter how fast tomorrow's conventional silicon-based computer can become, in order to solve specific classes of problems [especially nondeterministic polynomial (NP) problems], it may take the fastest silicon-based computer months or even years to process the calculations. This is mainly due to the serial computing nature of the conventional silicon-based computer. Therefore, searching out other possibilities to replace the current silicon-based computer catches the attention of researches from different fields. Among various approaches, DNA-based computing seems to be the most feasible way to solve NP problems.

In 1953, Watson and Crick proposed the double helix structure of DNA molecules which stated that each nucleotide consists of one of the four possible bases—adenine (A), thymine (T), guanine (G), and cytosine (C). Two nucleotides can form base pairs by following the complementary rules—A always pairs with T and G always pairs with C, which is known as Watson-Crick base pairing. This discovery is undoubtedly the cornerstone of DNA-based computing. It unveils that DNA molecules can be selected as information carrying medium. The most significant breakthrough in DNA-based computing was the successful demonstration of the DNA-based computing concept by using the laboratory method to solve a directed Hamiltonian path problem [1]. DNA-based computing can be implemented by two techniques [2]: solution (or test-tube) technique and surface technique. The solution technique was demonstrated to a satisfactory (SAT) problem [3] and other NP problems [4–6], while the surface technique was demonstrated to a bipartite maximum matching problem [7] and a Boolean logic circuit problem [8]. In this Letter, the strategic assignment problem, as a NP problem, is solved by using a newly proposed surface DNA-based computing technique.

Strategic assignment problems.—The strategic assignment problem is to find a maximum cardinality matching in a bipartite graph, in which each vertex in set X is matched with a nonrepeatable vertex in set Y. Such a problem is very closely related to strategic application. Sun Bin, an alleged descendant of Sun Tzu, author of the world famous strategy book known as ''The Art of War,'' successfully assisted Tian Ji in reversing the losing trend in a horse race through manipulating the sequence of houses selected in each round. This famous strategy application story is known as Tian Ji's horse racing [9]. The story about horse racing took place between two parties: Tian Ji and King Wei of Qi. There were a total of three rounds in one race, one of the three available horses for each party needed to be selected in each round, the party who won two out of three rounds won the race. According to the speed of the horses, each party's horse could be divided into three classes: regular, plus, and super. In the same class, those horses belonging to King Wei of Qi were slightly faster than those of Tian Ji. Tian Ji's horse racing problem is a special case of finding the maximum cardinality matching in a bipartite graph as illustrated in Fig. 1.

A bipartite graph is a graph G in which the complete vertex set is partitioned into two subsets, namely, X and Y. Every edge (i, j) in the graph G has one end in set X and the other in set Y. A matching in graph G is a set M (stands for matching) of edges of G such that no two of them have a vertex in common. The strategic assignment problem is as follows: given a graph containing two sets of vertices X, Y, and edge E, what is the maximum cardinality matching M consisting of the greatest number of edges?

In Fig. 2(a), graph G consists of a total of seven vertices, three in vertex set X and the remaining in vertex set Y, and five edges. From visual inspection, it is easy to figure out that the maximum cardinality matching can be either $M_1[(a, 4), (b, 1), (c, 2)]$ or $M_2[(a, 4), (b, 1), (c, 3)]$. The reason for selecting Fig. 2(a) as the study case is mainly because it is a general case of strategic assignment problems, whose number of vertices in vertex set X and Y are not equivalent as shown in Fig. 1. The selected case is relatively simple for demonstration purposes. However, the proposed

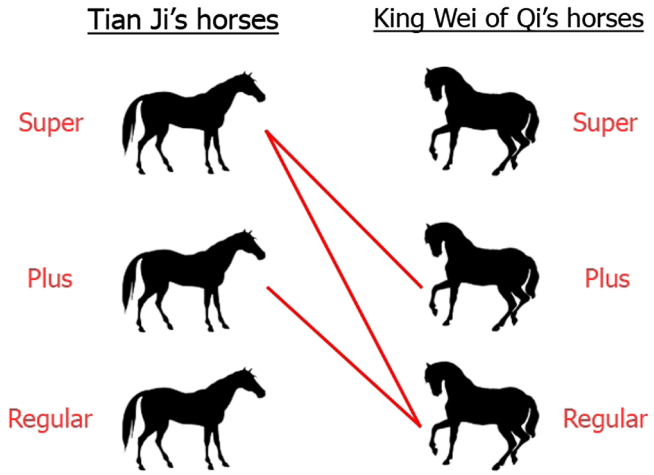

FIG. 1 (color online). Bipartite graph of Tian Ji's horse racing.

DNA-based computing method can be capable of scaling up to solve a more complicated strategic assignment problem.

Method.—The strategic assignment problem is a typical NP-complete problem. The required computational time increases exponentially corresponding to problematic size. The classical idea of solving the strategic assignment problem is to validate whether there exists an augmenting path P within a bipartite graph $G = (X, Y; E)$ with a given matching set M. If "yes," then the matching set M is not a maximum cardinality matching, modifying the given matching set M is required, and the whole process has to repeat until the augmenting path no longer exists in graph G. If "no," the matching set M is a maximum cardinality matching.

This classical idea theoretically demonstrates the procedures by which a conventional silicon-based computer solves strategic assignment problems. However, the classical idea itself is imperfect as it is both practically infeasible and fails to utilize the advantages of DNA-based computing.

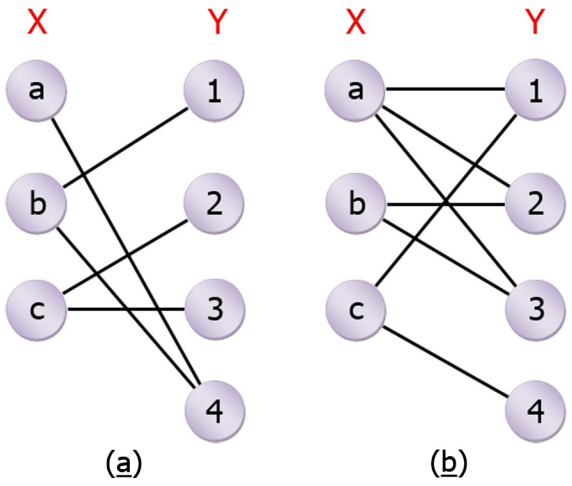

FIG. 2 (color online). Strategic assignment problem: (a) bipartite graph and (b) complementary graph, which connects vertices that are absent in (a).

First, the classical idea requires double-stranded DNA to be detached from the surface and to make a subsequent gel separation. Until now, there has been no existing laboratory technique or any chemical to collect efficiently the attached molecules from the surface, which controls the loss of information under an acceptable range. Nowadays, most DNA-based computing results are still illustrated in a fluorescence image or a histogram form. Second, the classical idea relies too much on human involvement. As mentioned earlier, the classical idea is designated to verify whether there is an augmenting path P present for a specified matching set M. In reality, it does not really solve strategic assignment problems. Most decisions are very subjective. Finally, the solutions at each stage of DNA-based computing cannot be recycled. In other words, if an augmenting path exists (i.e., the matching set M is not a maximum cardinality matching), the manual operation is required. This is specifically impractical once the computation starts. In order to overcome these limitations presented in the classical idea, the following DNA-based computing approach is proposed.

Implementation.—The following DNA-based algorithm is designed to solve the strategic assignment problem:

Step 1: Generate all possible solution paths through the graph.

Step 2: Retain solution paths begun with vertex in set X and ended with vertex stop.

Step 3: Retain solution paths with a specified length.

Step 4: Retain solution paths containing each vertex at most once.

Step 5: Determine the final solution path and the maximum cardinality matching set M.

DNA-based computing relies highly on how information is encoded to the DNA sequence. In the example, as illustrated in Fig. 2(a), there are a total of three vertices in set X—vertex a, b, and c. Each vertex in set X is encoded with 20 mer single-stranded DNA (ssDNA, or sometimes, oligonucleotide) with 50% CG (cytosine-guanine) content as described below, whereas the underlined sequences represent restriction enzyme sites corresponding to enzyme *Streptomyces caespitosus* (ScaI), *Streptomyces tubercidicus* (StuI), and *Serratia marcescens* (SmaI), respectively. These restriction enzymes work with different restriction and digestion sites and are only effective on double-stranded DNA (dsDNA) molecules. As a result of digestion, the original 20 base pair (bp) dsDNA is cut into two dsDNA of 10 bp length with blunt ends.

Each vertex in set Y is encoded with 20 mer single-stranded DNA with 50% CG content as stated below, whereas the underlined sequences represent the restriction enzyme sites corresponding to enzyme *Proteus vulgaris* (PvuII), *Escherichia coli* (EcoRI), *Haemophilus parainfluenzae* (HpaI), and *Pseudomonas maltophilia* (PmlI), respectively. The working principle of the selected restriction enzymes is exactly the same as previously applied in vertex sequences in set X.

Vertex	Single-stranded DNA sequence (5' to 3')
a	ATGCCGTAGTACTAAGCAGC
b	TATCGACAGGCCTATCGATC
c	TATTGTCCCCGGGGATCTAT
1	AACGTAGCAGCTGTTACGCT
2	TCTCTGAGAATTCCCGGCTA
3	CGCCTGTGTTAACGCGTAAT
4	GTACTTGCACGTGTAACGTG

There are two classes of edges, both treated as directed edges. In the first class, the edges (i, j), $i \in \{a, b, c\}$ and $j \in \{1, 2, 3, 4\}$, are illustrated in Fig. 2(a). These edges are used to connect from vertex i in set X to vertex j in set Y. For each edge (i, j) sequence, it contains two sections the first section, 10 mer ssDNA is the complementary strand of rear 10 mer ssDNA of vertex i in set X; the second section, 10 mer ssDNA is the complementary strand of former 10 mer ssDNA of vertex j in set Y.

In the second class, the edges (j, i) are the ones in the complementary of the original graph G as illustrated in Fig. 2(b). These directed edges are used to connect vertices in set Y to vertices in set X through backtracking. Therefore, this special class of edges is named backtracking edges. The objective of backtracking edges is used to establish a solution path. Similarly, for each backtracking edge (j, i), it contains two sections the first section, 10 mer ssDNA is the complementary strand of rear 10 mer ssDNA of vertex j in set Y; the second section, 10 mer ssDNA is the complementary strand of former 10 mer ssDNA of vertex i in set X.

In addition to all the vertices and edges described above, a special class of vertex, called stop, is added and used to regulate the solution path—no matter which vertex the path begins with, it should end with the stop vertex. These unqualified paths are eliminated. There are totally four stop vertices corresponding to four number of vertices in set Y, namely, stop 1, stop 2, stop 3, and stop 4, respectively. Each stop vertex is a 10 bp dsDNA with 10 mer sticky end. The 10 mer sticky end is designed as the complementary strand of rear 10 mer ssDNA of vertex j. These stop vertices are summarized below:

Vertex	Double-stranded DNA sequence
Stop 1	5'-ATGCGTACTG-3' 3'-GACAATGCGATACGCATGAC-5'
Stop 2	5'-ATGCGTACTG-3' 3'-AAGGGCCGAGTACGCATGAC-5'
Stop 3	5'-ATGCGTACTG-3' 3'-TTGCGCATTATACGCATGAC-5'
Stop 4	5'-ATGCGTACTG-3' 3'-CACATTGCACTACGCATGAC-5'

Step 1: After all DNA strands as stated above are synthesized in separate test tubes—X (vertex set X), Y (vertex set Y), E (edge), B (backtracking edge), and stop (S vertex), a small amount of the solution from every test tube is mixed in a new test tube to allow for hybridization. After hybridization, T4 DNA ligase is added to the solution and sufficient time is given to allow the completion of ligation. It is believed that all possible solution paths are generated in the new test tube (solution) as for each edge in the graph. There are approximately 6×10^{13} copies of the associated ssDNA. For convenience, each possible solution path can be expressed in the form of "vertex $\rightarrow$ vertex $\rightarrow \ldots \rightarrow$ vertex." For instance, the path, $a \rightarrow 4 \rightarrow b \rightarrow 1 \rightarrow c \rightarrow 2 \rightarrow$ stop 2, represents the DNA sequence illustrated in Fig. 3.

By applying such encoding theme, all possible solution paths are only limited to alternating paths, which consist alternately of edges in the strategic assignment problem and its complementary graph. This simply means that the paths containing two adjacent vertices from the same vertex set, for instance, $a \rightarrow 4 \rightarrow 2$ or $a \rightarrow b \rightarrow 1$, never appear in the possible solution path during hybridization.

Step 2: The product is amplified by polymerase chain reaction (PCR) using 10 mer complementary strand of former 10 mer oligonucleotide of vertex a in set X and 10 mer strand of stop vertex, which is fixed in sequence (3'-TACGCATGAC-5') as a primer. As the result of PCR, only solution path begun with any vertex in set X and ended with any stop vertex is amplified. The incomplete paths, for instance, "$b \rightarrow 1 \rightarrow c \rightarrow 3 \rightarrow a$," are eliminated.

Step 3: Two wells are built on one end of the gel slab. One well "O" is used to contain the product, while another well "F" is used to contain the DNA molecules with a known length. According to the working principle of gel electrophoresis, the DNA migration length towards the other end of the gel slab, as a result of electrophoresis, is inversely proportional to the length of DNA molecules. In this case, the desired DNA molecule should be exactly 130 bp (i.e., the desired solution path must contain exactly the six-vertex sequence, each with 20 bp, and a stop vertex with 10 bp). After completing the gel electrophoresis process, the solution in well O, which has exactly the same length of molecule in well F, is collected. In other words, any path having a length either longer or shorter than 130 bp, is discarded.

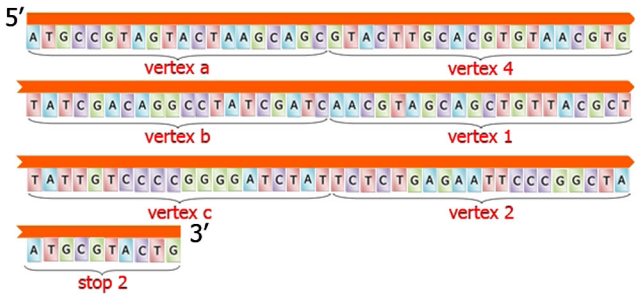

FIG. 3 (color online). Path DNA sequence.

Step 4: The specific restriction enzyme is applied to eliminate paths containing any vertex in graph G more than once. This can be achieved by destroying the solutions containing more than one restriction site. It is necessary to note that the most important is the information associated with the solution path, rather than detailed DNA sequence itself. For instance, two solution paths "$a \to 4 \to c \to 2 \to b \to 1$" and "$b \to 1 \to a \to 4 \to c \to 2$" deliver the same information, that is, the matching set $M[(a, 4), (b, 1), (c, 2)]$. In this case, the solution path starts from vertex a and ends with one of the stop vertices.

There are a total of five types of restriction enzymes used in this experiment—ScaI (a), PvuII (1), EcoRI (2), HpaI (3), and PmlI (4). The objective is to eliminate the paths passing through any vertex in graph G more than once. First, the restriction enzyme ScaI is added into the solution and given sufficient time to allow the completion of digestion. After that, the complete solution is split into four test tubes. Each test tube has added to it restriction enzymes PvuII, EcoRI, HpaI, and PmlI separately. These tubes are labeled corresponding to the added enzyme. Finally, all solutions from the test tube are inserted into a 5-well gel slab. Each well "1" to "4" contains the solution with four different enzymes, and well "5" contains the DNA molecules with known length—100 bp. Only the solution in any line of the gel slab with the same length as in the calibration line (well 5) is kept. It is necessary to understand how each procedure works. First, the restriction enzyme ScaI is added to the solution. The restriction enzyme ScaI digests the solution path containing the sequence of vertex a. As a result, "$a \to 4 \to c \to 2 \to a \to 4 \to S$" and "$a \to 4 \to c \to 3 \to a \to 4 \to S$" are eliminated from the final solution pool. The paths contain more than one sequence of vertex a. In other words, they contain more than one restriction site for enzyme ScaI. Therefore, as a result of applying restriction enzyme ScaI to the solution, the paths break into three portions of dsDNA—10, 40, and 80 bp. None of these portions of dsDNA satisfy the length requirement as specified by the experiment, which is 100 bp. By following the same idea, the paths "$a \to 4 \to c \to 2 \to b \to 4 \to S$" and "$a \to 4 \to c \to 3 \to b \to 4 \to S$" are eliminated by the combination of restriction enzymes EcoRI and HpaI. Only two paths "$a \to 4 \to c \to 2 \to b \to 1 \to S$" and "$a \to 4 \to c \to 3 \to b \to 1 \to S$" remain in the well. The restriction enzyme ScaI cut two path sequences into 10 and 120 bp, and the restriction enzyme PvuII cut the remaining 120 bp sequence into 100 and 20 bp dsDNA. In the subsequent gel electrophoresis, only these two paths satisfy the length requirement. Although half of the sequences of vertices a and 1 are destroyed as a result of the restriction enzyme, it does not affect our analysis of the DNA sequence in order to answer the strategic assignment problem.

Step 5: The product is extracted from the gel slab. DNA sequencing is required to determine the actual sequence within the solution. The sequencing can be done by either Sanger's method or an advanced method like real-time DNA sequencing from single polymerase molecules. Consequently, the DNA sequence can decipher the answer to the strategic assignment problem.

Conclusion.—The proposed DNA-based computing methodology has several major advantages over existing methods. First, it provides an instant result instead of manually verifying the existence of augmenting paths. Second, it is capable of scaling up as it requires the least amount of human involvement. Last, it is relatively simple and practicable. The minimal number of backtracking edges $e = \min(m, n) - 1$, where m and n are the number of vertices in sets X and Y, respectively. It is believed that the proposed approach of solving strategic assignment problems will have much potential impact on strategy-related applications in the future.

*Author to whom correspondence should be addressed.

[1] L. M. Adleman, Science **266**, 1021 (1994).
[2] Q. H. Liu, L. M. Wang, A. G. Frutos, A. E. Condon, R. M. Corn, and L. M. Smith, Nature (London) **403**, 175 (2000).
[3] R. J. Lipton, Science **268**, 542 (1995).
[4] Q. Ouyang, P. D. Kaplan, S. M. Liu, and A. Libchaber, Science **278**, 446 (1997).
[5] Z. X. Yin, F. Y. Zhang, and J. Xu, J. Chem. Inf. Comput. Sci. **42**, 222 (2002).
[6] F. S. Xiong, D. Spetzler, and W. D. Frasch, Integr. Biol. **1**, 275 (2009).
[7] S. Y. Wang, J. Math. Chem. **31**, 271 (2002).
[8] X. P. Su and L. M. Smith, Nucleic Acids Res. **32**, 3115 (2004).
[9] J.-J. Shu, Interdiscipl. Sci. Rev. **36**, 1 (2011).