An improved surface-based method for DNA computation

Haoyang Wu

Biochip Research and Development Center, Department of Biology Science and Biotechnology, Tsinghua University, Beijing 100084, People’s Republic of China

Accepted 28 August 2000

Abstract

DNA computing is a novel method for solving a class of intractable computational problems, in which the computing time can grow exponentially with problem size. Up to now, many accomplishments have been achieved to improve its performance and increase its reliability, among which a surface-based method is an efficient candidate. In this paper, the surface-based approach proposed by Liu, Q., Wang, L., Frutos, A.G., Condon, A.E., Corn, R.M., and Smith, L.M., 2000, DNA computing on surfaces. Nature 403, 175–179 is analyzed and an improved surface-based method for DNA computation (i.e. the hybrid DNA-optical computing method) is proposed. Compared with Liu et al.’s approach, our method has some significant advantages such as low cost, short operating time, reusable surface and simple experimental steps. Moreover, the concept of combining easily patterned DNA computing steps with equally parallel, but generally uniform and not easily patterned optical computing steps is an important new direction.

Keywords: DNA computing; Optical computing; NP-complete; Satisfiability problem

1. Introduction

Since Adleman (1994) demonstrated that a directed Hamiltonian path problem (HPP) could be encoded in DNA and evaluated in 1994, many researchers around the world have been attracted to a new field of DNA computation (Lipton, 1995; Frutos et al., 1997; Ouyang et al., 1997; Smith et al. 1998; Liu et al., 2000). The principle of DNA computation is not complicate. It uses mainly the molecular-biology operations of hybridization, ligation, cleavage with endonuclease and exonuclease, polymerase chain reaction (PCR) amplification, etc. The most attractive advantage of DNA computation is its massive parallelism. By using DNA-based computers, time required to solve hard problems [like the non-deterministic polynomial-time complete (NP complete) set of problems] may be limited in polynomial time, in contrast to exponential time needed by current silicon-based computers. Due to these exciting anticipations, DNA has been touted as a novel medium for computationally complex problems and DNA computation has become a focus of extensive speculation.

However, so far no large, computationally complex problem has yet been encoded in DNA and
solved by molecular biology methods. The DNA computations that have been carried out rely on a relatively small number of oligonucleotides that represent a correspondingly small number of either possible or correct sequence solutions. For example, the HPP was encoded with 20 oligonucleotide strings (Adleman, 1994), a maximal clique problem was encoded in 25 strings (Ouyang et al., 1997). The failure of DNA to rival silicon may be the result of two reasons — firstly DNA computation requires an exponentially increasing number of DNA molecules, but in practice, the demand for such large number of DNA strands is hard to meet, therefore, this requirement becomes a bottleneck for DNA computation. Secondly, the average error rate during the evaluation of DNA computation is not low enough, therefore, some factors such as incorrect hybridization, possible internal secondary structure in oligonucleotides may reduce the reliability of the final results. These two shortcomings restrict DNA computation from being scaled up to large computationally complex problems.

In order to overcome these two disadvantages, many papers have been reported, among which a surface-based method involving the immobilization and manipulation of combinatorial mixtures of DNA on a support is an effective approach (Frutos et al., 1997; Smith et al., 1998; Liu et al., 2000). In the sections that follow, we will discuss the current surface-based method briefly and then give an improved surface-based method for DNA computation.

2. DNA computation based on a surface

According to the literatures reported so far, the manners, in which complex combinatorial sets of DNA molecules may be manipulated can be classified as two formats — (i) in solution (solution phase format); or (ii) attached to a surface (solid-phase format). Smith et al. (1998) proposed that the solid-phase format possessed many important advantages over the solution-phase format, including facilitated sample handling, decreased losses during sample handling, reduction of interference between oligonucleotides, and facile purification of the DNA molecules at every step of the experiment.

Liu et al. (2000) proposed a surface-based DNA computation method for solving a satisfiability (SAT) problem. The SAT problem is an NP-complete problem in Boolean logic. An instance of the SAT problem consists of a set of Boolean logic variables separated by the logical OR operation (denoted by ‘∨’; \( u \lor v = 0 \) if and only if \( u = v = 0 \)) within clauses, and with the clauses separated by the logical AND operation (denoted by ‘∧’; \( u \land v = 1 \) if and only if \( u = v = 1 \)). The SAT problem is to find whether there are values for the variables that simultaneously satisfy each clause in a given instance of the problem. The SAT problem solved by Liu et al. (2000) is

\[
(w \lor x \lor y) \land (w \lor \neg y \lor z) \land (\neg x \lor y) \land (\neg w \lor \neg y)
\]  

where 4 variables \( w, x, y \) and \( z \) are employed (\( \neg x, \neg y, \neg z \) and \( \neg w \) are the negation of the variables \( w, x, y \) and \( z \); thus \( \neg x = 0 \) if and only if \( x = 1 \), and \( \neg x = 1 \) if and only if \( x = 0 \)). Each of the four variables can be either true (‘1’) or false (‘0’) and, thus, there are total of \( 2^4 \) candidate solutions.

![Fig. 1. Overview of the surface-based approach to DNA computations.](image)
Table 1
Relationship between oligonucleotides attached on the surface and values of variables

<table>
<thead>
<tr>
<th>Strand</th>
<th>S₀</th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
<th>S₅</th>
<th>S₆</th>
<th>S₇</th>
<th>S₈</th>
<th>S₉</th>
<th>S₁₀</th>
<th>S₁₁</th>
<th>S₁₂</th>
<th>S₁₃</th>
<th>S₁₄</th>
<th>S₁₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>wxyz</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
</tr>
</tbody>
</table>

The overall strategy for DNA computation on surface consists of five steps (Fig. 1). In the first two steps, a combinatorial set of single-stranded DNA molecules representing all possible solutions to a given computational problem is synthesized (‘make’) and immobilized in an unaddressed fashion (‘attach’) on a surface via a reactive function group X. For example, in Fig. 1, the total 16 oligonucleotides correspond to 16 candidate solutions, i.e. the oligonucleotide S₀ refers to the solution for \( w = 0, x = 0, y = 0 \) and \( z = 0 \), and S₁₅ refers to the solution for \( w = 1, x = 1, y = 1 \) and \( z = 1 \).

In each of \( N \) successive cycles of the DNA computation, subsets of the surface-bound combinatorial mixture are tagged by hybridization to their complements in a ‘mark’ operation. This operation is achieved by hybridizing to the surface those oligonucleotides that are complementary to the strands which do satisfy the clause. For example, for the first clause only two sequences do not satisfy this clause, namely, those for which \( w, x \) and \( y \) are set to zero (S₀[0000] and S₁[0001]), see Table 1). Thus in cycle 1, the complements (Cₘₙₜ) of the 14 other oligonucleotides (\( w = 1 \) (C₈, C₉, C₁₀, C₁₁, C₁₂, C₁₃, C₁₄, C₁₅); \( x = 1 \) (C₄, C₅, C₆, C₇, C₁₂, C₁₃, C₁₄, C₁₅); \( y = 1 \) (C₂, C₃, C₆, C₇, C₁₀, C₁₁, C₁₄, C₁₅)) are combined and hybridized to the surface. It can be seen that for the first variable \( w \) 2³ oligonucleotides (C₈, C₉, C₁₀, C₁₁, C₁₂, C₁₃, C₁₄, C₁₅) should be added to the surface; for the second variable \( x \) another 2² oligonucleotides (i.e. C₄, C₅, C₆, C₇) should be added, and for the third variable \( y \), another 2¹ oligonucleotides (i.e. C₂, C₃) should be added. Therefore, totally 14 (\( 1.75 \times 2^{n-1} \), \( n = 4 \)) oligonucleotides are required to be added to the surface in order to perform the ‘mark’ operation. The implementation of ‘mark’ operation will become more and more complicated as the number of variables increases.

After the ‘mark’ operation, an enzyme (e.g. *E. coli* exonuclease I) is added to destroy surface-bounded oligonucleotides present in an unhybridized single-stranded form (‘destroy’). Consequently, the surface is regenerated by removing all hybridized complements in an ‘unmark’ operation. Repetitive cycles of ‘mark’, ‘destroy’ and ‘unmark’ operations remove from the surface all strands, which do not satisfy the problem. At the end of \( N \) cycles, only those strands which are solutions to the problem will remain. Their identities are determined in a ‘read-out’ operation by PCR followed by hybridization to an addressed array.

Although Liu et al. claim that their surface-based approach is capable of scaling up to problems with more variables, it requires further research to make this approach really applicable. For example, by using Liu et al.’s approach, a 3-SAT problem with \( n \) variables will require at least \( 2^n \) unique DNA strands (3-SAT problem is a particular SAT problem in which there are just three variables in each clause), and at the same time each DNA strand should be designed carefully to avoid incorrectly hybridizing to other strands and their complementary strands. To synthesize such an exponentially increasing population of unique strands is a very hard task unaffordable to any user.

Another disadvantage of Liu et al.’s approach is that for a 3-SAT problem with \( m \) clauses and \( n \) variables, in order to perform the ‘mark’ operation in each cycle, one has to add \( 2^{n-1}, 2^{n-2}, 2^{n-3} \) oligonucleotides for the first, second and third variable of each clause respectively, which sum up to \( 1.75 \times 2^{n-1} \) oligonucleotides. Therefore, implementation of ‘mark’ operation will become more and more complicated as the number of variables increases.

These two disadvantages are serious bottlenecks that constrain the application of Liu et al.’s ap-
proach. In Section 3, an improved method is proposed to remove these bottlenecks, therefore, make the surface-based DNA computation more practical.

3. An improvement to Liu et al.’s approach

For a 3-SAT problem with \( m \) clauses and \( n \) variables \( x_1, x_2, \ldots, x_n \), we firstly synthesize \( 4n \) kinds of short oligonucleotides. The first \( 2n \) oligonucleotides represent \( x_1, x_2, \ldots, x_n \) and \( \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n \) respectively, and the rest \( 2n \) oligonucleotides each tagged with a fluorescent label are complementary to the former \( 2n \) strands, which are designated as \( x_1', x_2', \ldots, x_n' \) and \( \bar{x}_1', \bar{x}_2', \ldots, \bar{x}_n' \), respectively. In following discussions, the oligonucleotide \( x_1 \) means the strand corresponding to the variable \( x_1 \), and the oligonucleotide \( x_1' \) means the strand complementary to oligonucleotide \( x_1 \). It should be noticed that the \( 4n \) kinds of short oligonucleotides must differ from each other with at least 4 base mismatch in order to avoid incorrect hybridization among them.

Now consider a simple SAT problem with 3 variables \( x, y, \) and \( z \) — \( (x \lor y \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor z) \). We shall synthesize 12 kinds of short oligonucleotides, where six oligonucleotides represent \( x, y, z \) and \( \bar{x}, \bar{y}, \bar{z} \), the other six represent the complementary sequences of \( x, y, z \) and \( x, y, z \), which are designated as \( x', y', z' \) and \( \bar{x}', \bar{y}', \bar{z}' \). After the 12 kinds of short oligonucleotides are successfully synthesized, we then generate a set of long oligonucleotides by combinatorially concatenating different short oligonucleotides, i.e. the first segment of a long oligonucleotide is \( x \) or \( \bar{x} \), the second segment is \( y \) or \( \bar{y} \), and the third segment is \( z \) or \( \bar{z} \). As shown in Fig. 2, the set of all long oligonucleotides corresponding to the solution space.

In step 3 (‘mark’), for each literal of a certain clause, add the corresponding complementary strand to the surface. For example, for the first clause \((x \lor y \lor \bar{z})\), we should add oligonucleotides \( x', y' \) and \( \bar{z}' \) to the surface. Any solution, which satisfies this clause will be hybridized at least one complementary strand tagged with a fluorescent label. As shown in Fig. 3, after the ‘mark’ operation only the No.1 oligonucleotide is not marked.

In step 4 (‘detect’), the surface is detected using a method of fluorescence-image to determine whether there exist solutions that have bright intensity. A solution with bright intensity means that it satisfies the current clause. A high contrast black and white film is exposed in register with the surface. The film is then reversal processed into a positive slide, in which fluorescent locations on the surface yield clear spots on the otherwise black film.

In step 5 (‘unmark’), the temperature is raised to separate all double-stranded DNA into single-strands by thermal denaturation, thus, the surface is returned to the initial state shown in Fig. 2.

Step 3, 4, 5 constitute a cycle of operations. For a 3-SAT problem with \( m \) clauses, the cycle will be performed \( m \) times.
At last, we can determine whether the 3-SAT problem is satisfiable by detecting whether there exist solutions with bright intensity for all clauses. The processed films are placed in register with each other and the stack of films is illuminated from the back. The image of the stack is projected onto a photometer that integrates the light over the whole front of the film stack. If anywhere on the film area light can penetrate through all films, the photometer will show a response, which indicates that the SAT problem can be satisfied.

4. Conclusion

In this paper, an improved surface-based method for DNA computation is proposed. Compared with the approach reported by Liu et al. (2000), our improved method has some advantages as follows.

1. The number of unique oligonucleotides needed to be synthesized is reduced from \(2^n\) to \(4n\), where \(n\) is the number of variables. As a result, the cost is reduced too.

2. By using Liu et al.’s approach, one has to add \(1.75 \times 2^{n-1}\) kinds of oligonucleotides during the process of ‘mark’, whereas by using our method one needs only to add three kinds of oligonucleotides, hence the operating time is saved significantly.

3. By using Liu et al.’s approach, the surface can be used only once because at the end of experiment, only the solutions that satisfy the problem remain, and the others have been destroyed. Whereas by using our method, the surface is reusable, since after ‘unmark’ operation, the surface is regenerated as a new one.

4. By using Liu et al.’s approach, after all clauses have been disposed, one has to carry out some additional operations to identify the final result, i.e. PCR amplification followed by hybridization to an addressed array, whereas by using our method, the final result can be readily obtained. Moreover, the DNA computing steps proposed in this paper transform the SAT problem in such a way that it can be solved conveniently by the optical computing steps. Stacking the \(m\) films computes optically in parallel the \((m-1)2^n\) AND operations. Integrating the light over the face of the film stack computes in parallel \((2^n-1)\) OR operations among the \(2^n\) locations on the film.

References


