Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 633–637
Progress in Natural Science
www.elsevier.com/locate/pnsc

Special report

DNA computing model based on lab-on-a-chip and its application on solving the timetabling problem

Fengyue Zhang [a,*,1], Bo Liu [b,1], Wenbin Liu [c], Qiang Zhang [d]

[a] Department of Biomedical Engineering, School of Life Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[b] Chinese Academy of Inspection and Quarantine, Beijing 100029, China
[c] College of Computer Science and Engineering, Wenzhou University, Wenzhou, Zhejiang 325027, China
[d] Liaoning Key Laboratory of Intelligent Information Processing, Dalian University, Dalian, Liangning 116622, China

Received 21 December 2007; received in revised form 16 January 2008; accepted 17 January 2008

Abstract

The essential characteristic of DNA computation is its massive parallelism in obtaining and managing information. With the development of molecular biology technique, the field of DNA computation has made a great progress. By using an advanced biochip technique, laboratory-on-a-chip, a new DNA computing model is presented in the paper to solve a simple timetabling problem, which is a special version of the optimization problems. It also plays an important role in education and other industries. With a simulated biological experiment, the result suggested that DNA computation with lab-on-a-chip has the potential to solve a real complex timetabling problem.

Keywords: DNA computing; Timetabling problem; Lab-on-a-chip

1. Introduction

Since Adleman demonstrated the possibility of solving NP-complete problems by encoding a computer style problem in DNA sequences [1], the DNA computation has become a new vista of computation that bridged computer science and molecular biology. DNA computing provided a massive computational parallelism that allowed us to tackle intractable combinatorial problems exhaustively, in contrast to the exponentially increasing time required by a Turing machine. After Adleman's work, the field of DNA computation has made a great progress, which included mainly some theoretical studies [2–4], solution-based DNA computation [1,5–7] and surface-based DNA computation [8–14]. Generally, the surface-based DNA computation, which manipulates DNA strands immobilized on a surface, usually uses more advanced biological technology and has a potential to be automatized in contrast with the solution-based DNA computing.

As a very promising area of research, the focus of DNA computation is whether it could solve a hard arithmetic (like NP-hard) problem in reality. However, the most complex problem ever solved by DNA computation was a 20-Variable 3-SAT problem by Braich [15]. It was recognized that powering the theoretical computing capacity in a large reaction vessel was meaningless unless a reaction achieved the expected level of performance. Therefore, based on lab-on-a-chip which simply defined as a chip integrated with heaters, valves, pumps, microfluidic controllers, electrochemical and electroluminescent detectors [16–21], this paper proposed a DNA computing model in theory to solve a simple timetabling problem, which is a special version of the optimization problems found in real life situations but is not studied by DNA computation.

* Corresponding author. Tel./fax: +86 010 68914941.
E-mail address: zfywuhan@mail.hust.edu.cn (F.Y. Zhang).
[1] These authors contributed equally to this work.

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2008.01.007

2. DNA computing model based on lab-on-a-chip to solve a simple timetable

As a special version of the optimization problems found in real life situations, the timetabling problem, which plays an important role typically in education, has been periodically faced by every schools, colleges and universities all over the world. Like other optimization problems, the problem is computationally NP-hard. General timetabling problem could be defined as the scheduling of a set of lectures to which several groups of students must attend over a preset period of time, using some resources and satisfying a certain set of constraints [22].

A lot of studies have been carried out on timetabling problem [22–26]. Nowadays, this problem is still studied due to its variety and its complexity.

A simple timetable problem could be described as follows: a school with m teachers $X_1, X_2, \ldots, X_m$ and n classes $Y_1, Y_2, \ldots, Y_n$, known definitely that teacher X_i needs to have p_{ij} lectures to class Y_j, scheduling a set of lectures with as less hours as possible.

The DNA computing model can be described as follows:

Step 1. Encoding single-strand DNA chain to represent the class.
Step 2. Designing a lab-on-a-chip to be a logical calculator.
Step 3. Computing by adding single-strand DNA chain to the chip, if the value of logical calculation is true (i.e. satisfying the hard and soft constraints), keep the value, or delete the value.
Step 4. Repeating the DNA computing process mentioned above until all known lectures p_{ij} are obtained. The needed number of cycles would be the less class hours.

A simple example is given to describe the model in details: three teachers X_1, X_2 and X_3, four classes Y_1, Y_2, Y_3 and Y_4, the required teaching array $P = [p_{ij}]$ is

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

The hard constraints are

C_0 – A teacher could only give one lecture at a time.
C_1 – A room could only host one lecture at a time.
C_2 – A class could only attend one lecture at a time.
C_3 – Room capacities that Y_1 and Y_4 could attend lecture only in room a are respected.
C_4 – Teacher X_3 having a lecture to class Y_2 do so must on the second class hour.

The soft constraints are

S_0 – Enough rooms could be used for lectures.
S_1 – Class attending lecture should obey certain order that Y_j preceded Y_{j+1}.

A DNA computing model was proposed according to this special timetabling problem:

Step 1. Single-stranded DNA chains of Y_1, Y_2, Y_3 and Y_4 representing four classes were encoded and constructed as well as their complementary chains of $\overline{Y}_1$, $\overline{Y}_2$, $\overline{Y}_3$ and $\overline{Y}_4$. The designed DNA chains, not the complementary chains, were fixed on surface as probes, in which each designed single-stranded DNA chain included two sub-regions (Y_j or $\overline{Y}_j$ and E or $\overline{E}$), representing the certain class (Y_j) and its complementary strand ($\overline{Y}_j$) as shown in Fig. 1(a) and (b), respectively. A restriction enzyme site of $E/\overline{E}$ was designed at the end of double chains of $Y_j/\overline{Y}_j$, and the 5′ end of the $\overline{Y}_j$ chain was labeled with fluorescence which would be a positive reaction when activated by laser as shown in Fig. 1(c). According to the Watson–Crick principle, the complementary chains would form duplex that could be cut away from the surface in the restriction enzyme site if restriction enzyme reaction was operated as shown in Fig. 1(d). Therefore, the biological characteristic of DNA strands designed in this step would satisfy the hard constraints C_0, C_1 and C_2.

Step 2. The lab-on-a-chip was designed and it included three pools of hybridization representing three teachers X_1, X_2 and X_3. According to the required teaching array, single-stranded DNA chains were sited on the surface of the pools in which Y_1, Y_3 and Y_4 were sited on pool of X_1; Y_1 and Y_4 were on pool of X_2; and Y_2, Y_3 and Y_4 were on pool of X_3 as shown in Fig. 2(a).

Adding $\overline{Y}_1$ to the chip while closing all channels except those to X_1. $\overline{Y}_1/Y_1$ would form duplex region. After the pool X_1 was washed by buffer in a certain experimental condition, the fluorescence labeled on $\overline{Y}_1$ was a positive reaction when activated by laser as shown in Fig. 2(b). So the value of X_1Y_1 was true. Then close all channels of X_1 and go on to the next operation by adding $\overline{Y}_2$. $\overline{Y}_2$ would go to X_2 directly since the channels to X_1 had been closed. The result is illustrated in Fig. 2(c), in which X_2 is a negative reaction and X_3 is a positive reaction, i.e. X_3 Y_2 has true value. Repeating the operation by adding $\overline{Y}_3$ to the chip, there is a negative reaction in pool of X_2, which is a positive reaction by adding $\overline{Y}_4$, as shown in Fig. 2(d). In this step, the DNA computing model would satisfy the hard constraints C_0, C_1, C_2, C_3 and C_4 as well as the soft constraints S_0 and S_1 by controlling the adding order of

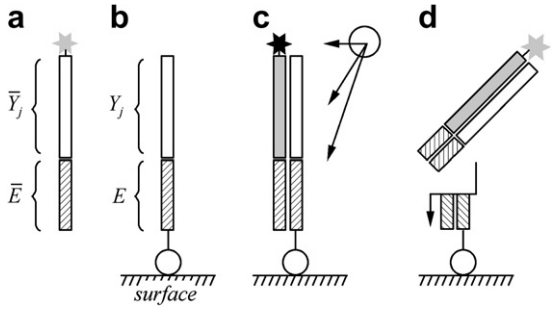

Fig. 1. The designed DNA strands on surface.