Available online at www.sciencedirect.com
ScienceDirect
BioSystems 91 (2008) 117–125
BioSystems
www.elsevier.com/locate/biosystems

Solving the SAT problem using a DNA computing algorithm based on ligase chain reaction

Xiaolong Wang *, Zhenmin Bao, Jingjie Hu, Shi Wang, Aibin Zhan

Department of Biotechnology, Ocean University of China, Qingdao 266003, People's Republic of China

Received 4 April 2007; received in revised form 12 August 2007; accepted 17 August 2007

Abstract

A new DNA computing algorithm based on a ligase chain reaction is demonstrated to solve an SAT problem. The proposed DNA algorithm can solve an n-variable m-clause SAT problem in m steps and the computation time required is $O(3m+n)$. Instead of generating the full-solution DNA library, we start with an empty test tube and then generate solutions that partially satisfy the SAT formula. These partial solutions are then extended step by step by the ligation of new variables using *Taq* DNA ligase. Correct strands are *amplified* and false strands are *pruned* by a *ligase chain reaction* (LCR) as soon as they fail to satisfy the conditions. If we score and sort the clauses, we can use this algorithm to markedly reduce the number of DNA strands required throughout the computing process. In a computer simulation, the maximum number of DNA strands required was $2^{0.48n}$ when $n=50$, and the exponent ratio varied inversely with the number of variables n and the clause/variable ratio m/n. This algorithm is highly space-efficient and error-tolerant compared to conventional brute-force searching, and thus can be scaled-up to solve large and hard SAT problems.

Keywords: DNA computing; SAT problem; Ligase chain reaction; Space complexity; Time complexity

1. Introduction

DNA computing is a newly emerging interdisciplinary science that uses molecular biotechnologies to solve problems in computer science or mathematics. In their pioneering studies, Adleman and Lipton solved combinatorial problems [Hamilton path problem (HPP) (Adleman, 1994) and satisfiability problem (SAT) (Lipton, 1995)], using DNA computing algorithms based on a brute-force search. At the beginning of computation, they constructed a DNA pool that contained the full-solution space, and then extracted correct answers and/or eliminated false ones from the pool step by step. Thus, the number of distinct DNA strands required in the initial data pool grows exponentially with the size of the problem, and eventually swamps the DNA data storage for large problems, which makes molecular computation impractical from the outset. Generally, it is believed that DNA computers that use a brute-force search algorithm are limited to 60 to 70 variables (Lipton, 1995). Recently, a few new algorithms, such as the breadth-first search algorithm (Yoshida and Suyama, 2000), and random walking algorithm (Liu et al., 2005), have been proposed. With the breadth-first search algorithm, the capacity of a DNA computer can be theoretically increased to about 120 variables (Yoshida and Suyama, 2000).

In the present study, we developed a new space-efficient DNA computing algorithm based on a *ligase chain reaction* (*LCR*), which uses only four operations: *ligating*, *amplifying*, *splitting* and *merging*. Instead of generating the full DNA library at the beginning,

* Corresponding author. Tel.: +86 532 82031970.
E-mail address: xiaolong@ouc.edu.cn (X. Wang).

0303-2647/$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2007.08.006

we start with an empty test tube, and generate partial solutions that only satisfy the clauses. The partial solutions are extended through *ligation* of new variable DNA: correct solutions are selectively *amplified* and false ones are *pruned* by *LCR*.

2. DNA Code Design

The SAT problem is an NP-hard computational problem that requires an exponential amount of time to solve with known sequential algorithms. Since all NP-

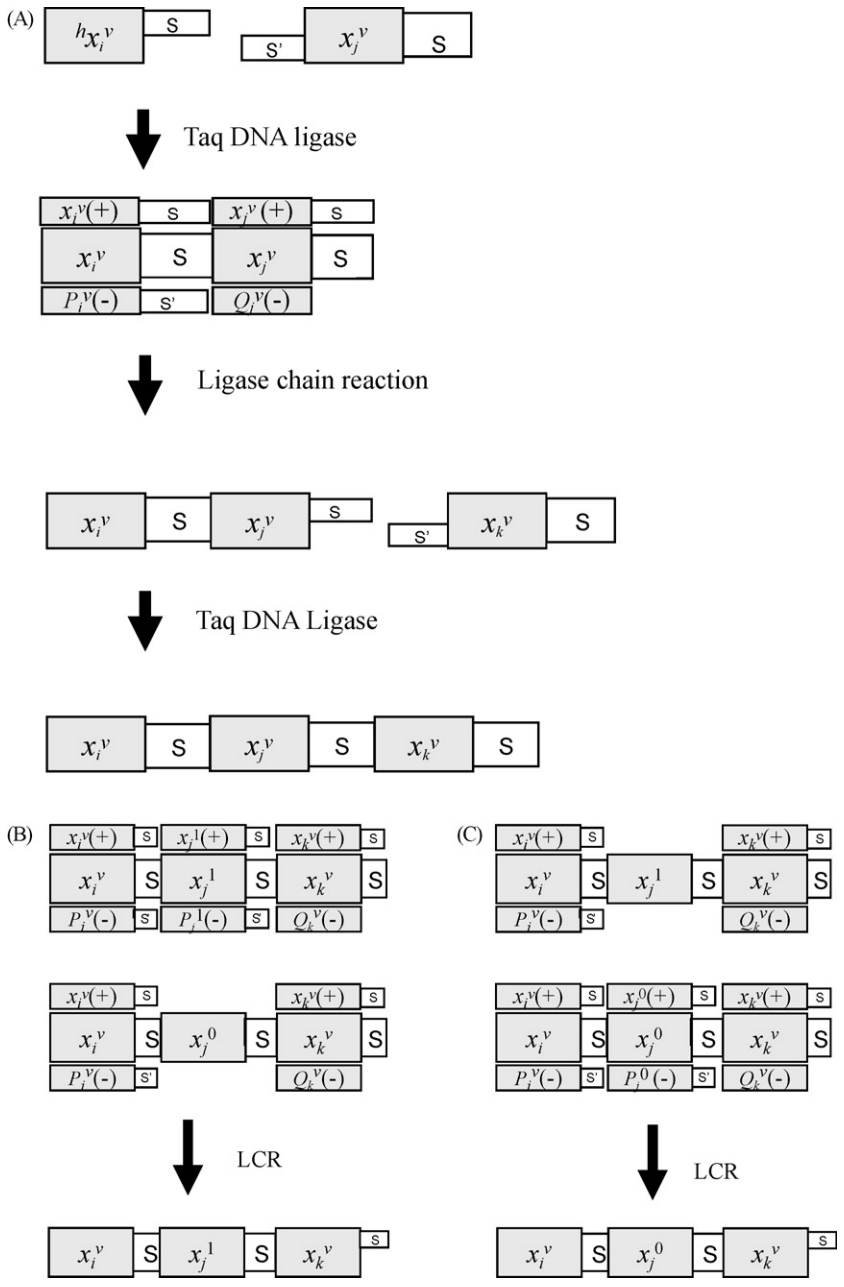

Fig. 1. The principle of the proposed DNA computer. (A) *Ligate* x_i^v with x_j^v to form $x_i^v - x_j^v$, *amplify* the ligation product by *LCR*, and then *ligate* x_k^v with $x_i^v - x_j^v$ to form $x_i^v - x_j^v - x_k^v$. Note that *s* and *s*' are respectively the sense and antisense strands of the linker sequence *S* (5′-ACTTTCCC-3′). (B) Computation of the literal x_j: DNA strands that contain x_j^1 (x_j^0) are (are not) amplified. (C) Computation of the literals $\sim x_j^v$: DNA strands that contain x_j^0 (x_j^1) are (are not) amplified.