Engineering 4 (2018) 61–77

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Cybersecurity—Article

A DNA Computing Model for the Graph Vertex Coloring Problem Based on a Probe Graph

Jin Xu [a,*], Xiaoli Qiang [b], Kai Zhang [c], Cheng Zhang [a], Jing Yang [d]

[a] Key Laboratory of High Confidence Software Technologies of Ministry of Education, Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[b] Institute of Novel Computer Science and Intelligent Software, Guangzhou University, Guangzhou 510006, China
[c] School of Computer Science, Wuhan University of Science and Technology, Wuhan 430081, China
[d] School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

ARTICLE INFO

Article history:
Received 10 December 2017
Revised 2 January 2018
Accepted 7 January 2018
Available online 25 February 2018

Keywords:
DNA computing
Graph vertex coloring problem
Polymerase chain reaction

ABSTRACT

The biggest bottleneck in DNA computing is exponential explosion, in which the DNA molecules used as data in information processing grow exponentially with an increase of problem size. To overcome this bottleneck and improve the processing speed, we propose a DNA computing model to solve the graph vertex coloring problem. The main points of the model are as follows: ① The exponential explosion problem is solved by dividing subgraphs, reducing the vertex colors without losing the solutions, and ordering the vertices in subgraphs; and ② the bio-operation times are reduced considerably by a designed parallel polymerase chain reaction (PCR) technology that dramatically improves the processing speed. In this article, a 3-colorable graph with 61 vertices is used to illustrate the capability of the DNA computing model. The experiment showed that not only are all the solutions of the graph found, but also more than 99% of false solutions are deleted when the initial solution space is constructed. The powerful computational capability of the model was based on specific reactions among the large number of nanoscale oligonucleotide strands. All these tiny strands are operated by DNA self-assembly and parallel PCR. After thousands of accurate PCR operations, the solutions were found by recognizing, splicing, and assembling. We also prove that the searching capability of this model is up to $O(3^{59})$. By means of an exhaustive search, it would take more than 896 000 years for an electronic computer (5×10^{14} s^{-1}) to achieve this enormous task. This searching capability is the largest among both the electronic and non-electronic computers that have been developed since the DNA computing model was proposed by Adleman's research group in 2002 (with a searching capability of $O(2^{20})$).

1. Introduction

In 1961, Feynman [1] proposed the idea of molecular computing, which was demonstrated by Adleman 33 years later. In 1994, Adleman [2] presented a DNA computing model in which DNA molecules were used as data, and enzymes and biological operations were used as tools in information processing. Since then, the DNA computing model has been developed in several new dimensions including theory, experiments, and applications. Studies in the past decade have shown that the DNA computing model is superior in solving optimal and graphical problems, especially in non-deterministic polynomial-time (NP)-complete problems. However, the main challenge of this model is that the size of DNA molecules used as data in information processing grows exponentially with an increase of problem size. This phenomenon is called exponential explosion, and is the biggest bottleneck preventing the development of DNA computing. To solve this problem, a parallel type of DNA computing model is proposed in this article, and several methods are used to overcome the biggest bottleneck.

In this section, we briefly introduce the latest developments in DNA computing research, and then outline the major innovative points of this article.

1.1. Previous results

In 1994, Adleman [2] first explored the computing feasibility of DNA molecules by presenting a DNA computing model for the

* Corresponding author.
E-mail address: jxu@pku.edu.cn (J. Xu).

https://doi.org/10.1016/j.eng.2018.02.011

Hamiltonian path problem. After that, many studies were designed to show the advantage of the huge parallelism that is inherent in DNA-based computing. In 1995, Lipton [3] proposed a DNA computing model to solve the satisfiability (SAT) problem. In 1997, Ouyang et al. [4] designed a DNA computing model for the maximal clique problem. By using the DNA hairpin formation, Sakamoto et al. [5] solved the SAT problem. Rothemund [6] attempted to use DNA as instruments to implement Turing machine. In addition, DNA computing was applied to manipulate gene expression [7,8]. Winfree [9] proposed a sticker model by DNA self-assembly. In 2002, Adleman's research group invented a DNA computer to solve a 20-variable 3-SAT problem with a searching ability of 2^{20} [10].

Graph vertex coloring is a way of coloring the vertices of graph G such that no two adjacent vertices share the same color. Although it is a classical NP-hard problem, graph coloring arises naturally in a variety of applications such as timetable schedules [11], register allocation [12,13], and so forth. The analysis of approximation algorithms for graph coloring started with the work of Johnson [14]. Subsequently, $O(n^{2/5}\log^{8/5}n)$ colors were used to color a 3-colorable graph with n vertices by Blum [15]. Karger et al. [16] provided a randomized polynomial time algorithm that colors a 3-colorable graph on vertices with $\min\left\{O(\Delta^{1/3}\log^{1/2}\Delta\log n),\ O(n^{1/4}\log^{1/2}n)\right\}$ colors. Schiermeyer [17] gave a complicated algorithm for deciding 3-colorability in less than time $O(1.415^n)$. Beigel and Eppstein [18] improved the bound by giving a fast algorithm for (3, 2)-constraint satisfaction problem (CSP) for solving the 3-coloring problem in time $O(1.3289^n)$.

The major research into graph vertex coloring based on DNA computing is as follows: In 1999, Jonoska et al. [19] proposed the potential application of three-dimensional (3D) structures in the field of DNA computing, and showed that some DNA structures could theoretically be used in DNA computing. In 2003, they constructed 3-arm and 4-arm DNA molecules to solve the graph vertex 3-coloring problem based on the aforementioned studies [20,21]. Liu et al. [22] proposed a DNA algorithm for the graph coloring problem based on surfaces. In 2003, Gao and Xu [23] presented a new DNA computing model for solving the graph vertex coloring problem, in which the technology of enzyme cutting was used to eliminate false solutions. In 2006, using the magnetic beads separation technology, Xu et al. [24] established a DNA computing model to solve the graph vertex coloring problem; this model was validated by a series of experiments.

1.2. Our results

We now present a novel DNA computing model for the graph vertex coloring problem; although we focus on analyzing $k = 3$, this method can be generalized to the situation of $k > 3$.

The basic idea of our model is to reduce the initial solution space by using the optimal method and process the data by a parallel polymerase chain reaction (PCR) operation. First, a given graph is divided into several subgraphs in order to make the bio-operations easier and delete as many false solutions possible. The subgraphs can then be solved in parallel according to the following steps: Determine the order of the vertices, determine the color set of each vertex in the subgraph, encode the DNA sequences, determine the probes, construct the initial solution space, and delete the false solutions (see Sections 3.3–3.7). Finally, the subgraphs are combined into the graph and implemented to delete false solutions gradually (see Section 3.8).

The main points of this article are as follows: ① The exponential explosion problem is solved by dividing subgraphs, reducing the vertex colors without losing solutions, and ordering the vertices in subgraphs; ② the bio-operation times can be greatly reduced by using a parallel PCR technology. Thus, the processing speed is remarkably improved in this model.

A 3-colorable graph with 61 vertices (Fig. 1) is used as an example to demonstrate how to use this computing model to solve the graph vertex coloring problem. In general, for a 3-colorable graph, the computing complexity is 3^n. Therefore, the computing capacity of our parallel DNA computing model can reach $O(3^{59})$, when the colors of vertices v_1 and v_n are given.

1.3. Outline of the article

This article is laid out as follows. In Section 2, we explain some notations and definitions of the graph coloring problem and PCR technology. In Section 3, the DNA computing model is illustrated, including algorithm steps, bio-operations, and technologies. In Section 4, a 3-coloring problem of a graph with 61 vertices is then solved by this computing model, and the specific experimental operations are described. Section 5 contains theory analysis, as we carefully examine the complexity of overcoming the exponential explosion phenomenon. In the last section, we conclude the article and point out the next possible research direction.

2. Notation and definition

2.1. The graph coloring problem

The work in this paper is always limited to finite, simple, and undirected graphs. In a given graph G, $V(G)$, $E(G)$, $d_G(v)$, and $\Gamma_G(v)$ denote the vertex set, edges set, degree of vertex v, and set of vertices adjacent to v of graph G, respectively, and are denoted using the short forms V, E, d_G, and Γ_G, respectively. We let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of a graph G, and denote the degree of v_i as $d(v_i)$ (abbreviated as d_i, $i = 1, 2, \ldots, n$). A walk denoted by W is an alternating sequence of vertices and edges, beginning and ending with a vertex, respectively, where each vertex is incident to both the edge that precedes it and the edge that follows it in the sequence, and where the vertices that precede and follow an edge are the end of that edge. A sequence of vertices is denoted by W, where $W = v_1 \ldots v_k$ ($k \geqslant 0$), beginning with vertex v_1 and ending with vertex v_k, and where v_i and v_{i+1} are adjacent ($i = 1, 2, \ldots, k-1$).

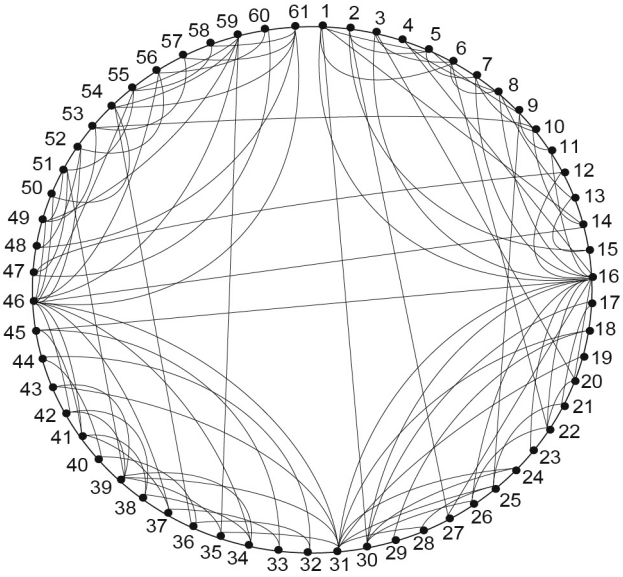

Fig. 1. A 3-colorable graph G with 61 vertices.