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 ($i = 1, 2, \ldots, k-1$).

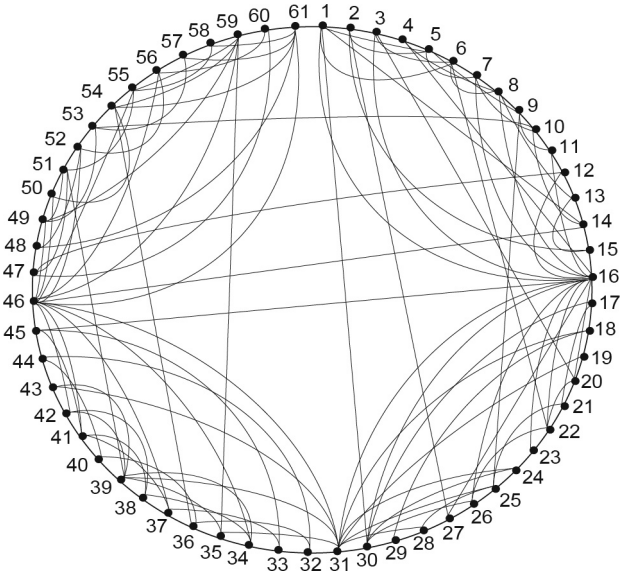

Fig. 1. A 3-colorable graph G with 61 vertices.

The length of a walk is the number of edges in it. A path is a walk in which all the vertices are distinct.

The vertex coloring of a graph G is an assignment from its vertex set to a color set such that two ends of each edge are assigned two different colors. Namely, it is a partition of V, where $V(G) = V_1 \cup V_2 \cup \cdots \cup V_k$, $V_i \neq \varnothing$ (empty set), and $V_i \cap V_j = \varnothing$, $i = 1, 2, \ldots, k$.

The minimum number of colors needed for such a coloring is called the chromatic number of G, and is usually denoted by $\chi(G)$. If a graph G can be colored by k ($k \geq \chi(G)$) colors, then the color set is denoted by $C_k(G)$. In this article, we limit ourselves to a 3-colorable graph G and we always assume that $C_3(G) = \{r, b, y\}$, where r, b, and y denote red, blue, and yellow, respectively. In other words, the 3-coloring problem of a given graph G can be solved if we find a map f: $V(G) \to \{r, b, y\}$ with the property $\forall uv \in E(G)$, $f(u) \neq f(v)$. The 3-coloring problem is a classical NP-complete problem [25]. Any terms and notations not defined here can be found in Ref. [26].

Definition 2.1: Let G be a simple graph with the vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$. Let $v_{i1}v_{i2}\ldots v_{in}$ be an ordering of $V(G)$, where $v_{ij} \in V(G)$, $j = 1, 2, \ldots, n$, and $v_{ij} \neq v_{il}$ if $j \neq l$. The number of edges in the ordering $v_{i1}v_{i2}\ldots v_{in}$, denoted by $N_{\text{edge}}(v_{i1}v_{i2}\ldots v_{in})$, is defined as follows:

$$N_{\text{edge}}(v_{i1}v_{i2}\ldots v_{in}) = \left|\left\{v_{ij}v_{i(j+1)} \in E(G);\ j = 1, 2, \ldots, n-1\right\}\right| \quad (1)$$

2.2. Polymerase chain reaction

PCR is a technique that amplifies a few copies of DNA across several orders of magnitude, thereby generating millions or more copies of a particular DNA sequence. The three parts of PCR are carried out in the same vial with different temperatures. The three steps in PCR are: the denaturation of the strands, annealing the primer to the template, and the extension of new strands. At the end of a cycle, each DNA template in the vial has been exponentially duplicated. After n cycles, the number of the DNA molecules is $y = (1 + x)^n$, where y is the copy number of DNA molecules, x is the efficiency of amplification, and n is the number of cycles.

3. DNA computing model

In this section, we describe the DNA computing model in detail, including algorithm steps, bio-operations, and technologies.

3.1. The algorithm of the model

In this model, limited PCR operations are exploited to delete the false solutions of a graph, and the DNA strands representing the true solutions of a given graph are sequenced to obtain the coloring of the given graph. The algorithm of this model is shown in Fig. 2.

3.2. Subgraph division and bridge vertices determination

Subgraph division can improve the computing efficiency in two aspects: On the one hand, the experiment can be operated more easily; on the other hand, most of false solutions can be deleted when constructing the initial solution space. Therefore, the appropriate division of a given graph G is of great significance in our model. Here G_j, $V(G_j) \subset V(G)$, $E(G_j) \subset E(G)$, where $j = 1, 2, \ldots, m-1$, is used to denote the primary subgraph. The method of subgraph division is demonstrated as follows:

Step 1: Determine the size of the subgraphs. The number of vertices in each subgraph should be consistent, ranging from 15 to 20. In addition, the edges in a subgraph should be as numerous as possible. In this way, the experiments can be conveniently performed, and most of the false solutions will be deleted when the initial solution space is constructed for each subgraph. Taking the graph G in Fig. 3 as an example, there are two divisions. In one case, the graph can be divided into two subgraphs G'_1 with vertex set $V'_1 = \{1, 2, 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 22\}$, and G'_2 with $V'_2 = \{8, 9, 10, 11, 12, 13, 14, 15, 23, 24, 25, 26, 27, 28, 29, 30\}$. In another case, the graph can be divided into two subgraphs G'_3 with $V'_3 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ and G'_4 with $V'_4 = \{16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30\}$. Obviously, the latter division method is better because the number of edges in subgraphs G'_3 and G'_4 is greater than in G'_1 and G'_2.

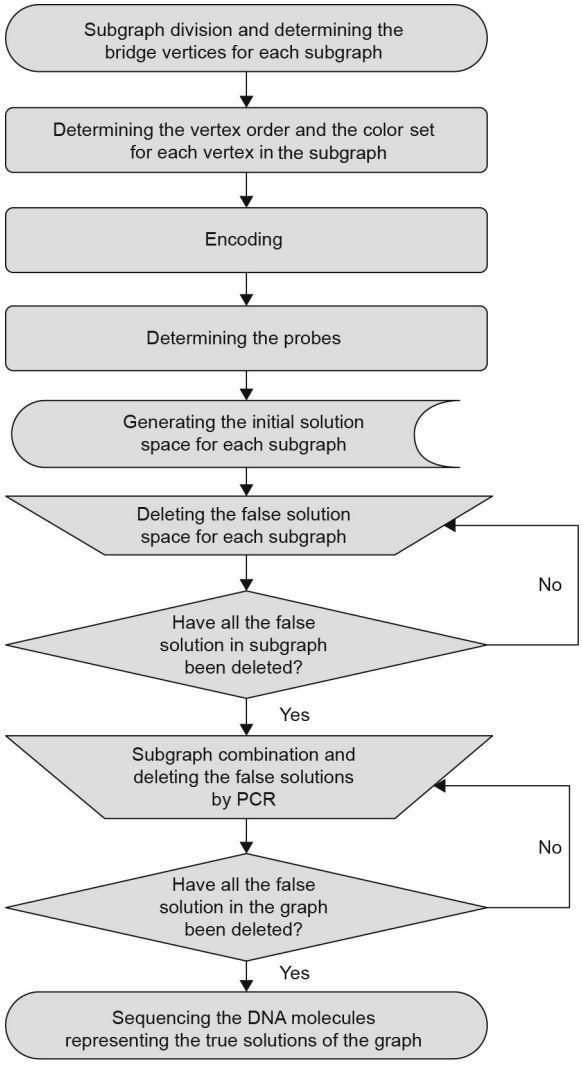

Fig. 2. A flowchart of the algorithm.

Step 2: Obtain the first subgraph G'_1 with the bridge vertices u_1 and u_2. The bridge vertices should be adjacent and the sum of the degree of the bridge vertices should be maximal. If more than one pair of bridge vertices satisfy these conditions, we select the one pair (u_1 and u_2) that result in $N_{\text{edge}}(u_1 = v_{i1}v_{i2}\ldots v_{in} = u_2)$ being the greatest (see Section 3.3 for more details). If more than one pair of bridge vertices satisfy this condition, we then choose any pair.

Step 3: Determine the second subgraph G'_2 containing bridge vertiex u_2 and a new bridge vertex u_3, which can be found using the following methods: The bridge vertex should be adjacent to the vertex u_2, and the degree is maximal in G'_2. If more than one pair of bridge vertices satisfy these conditions, then choose one pair according to Step 2.

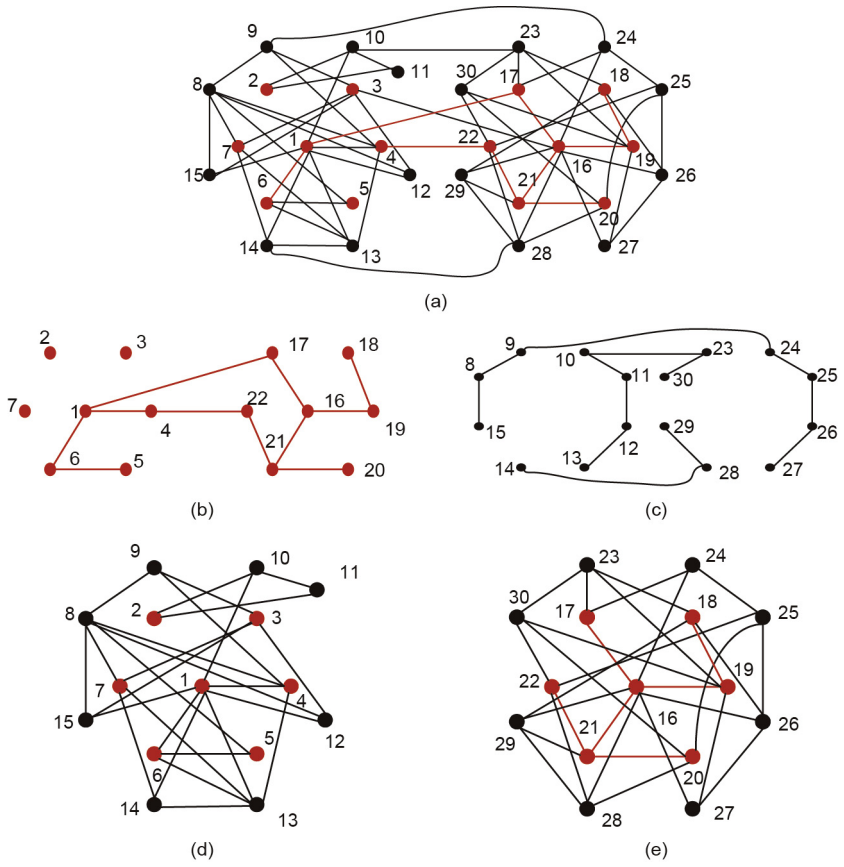

Fig. 3. An example of primary subgraph division. (a) Graph G; (b) subgraph G'_1; (c) subgraph G'_2; (d) subgraph G'_3; (e) subgraph G'_4.

Step 4: Perform similar operations as those in Step 3 to obtain the other subgraphs with their bridge vertices.

It should be noted that: ① the size of the vertex set of the last subgraph G_m may not be equal to the others; and ② the bridge vertices in the last subgraph G_m may be u_1 and u_{m-1}, which is the bridge vertex of G_{m-1} if u_1 and u_{m-1} are adjacent.

In our DNA computing model, the bridge vertices are very important for the subgraph combination. The principle in choosing bridge vertices is that the sum of the degrees of two bridge vertices in one subgraph should be as large as possible. In this way, the size of the color set of the vertices adjacent to bridge vertices can be reduced (see Section 3.3). Moreover, the size of the initial solution space can also be reduced.

According to the steps above, the four divided subgraphs of graph G in Fig. 1 are shown in Fig. 4; the bridge vertices are 1, 16, 31, 46, and 61, respectively.

3.3. Determining the order and color set for each vertex in the subgraphs

After subgraph division, the bridge vertices in each subgraph are determined. The algorithm for ordering the vertices in a subgraph should satisfy the following conditions: ① The bridge vertices must be considered as two endpoints in the ordering of the vertices subgraphs; and ② the number of edges in ordering (see Definition 2.1) should be as great as possible. Here, we take subgraph G_1 as an example. Suppose that the vertex set of subgraph G_1 is $V(G_1) = \{v_1, v_2, \ldots, v_t\}$ with the bridge vertices v_1 and v_t. Without loss of generality, let the vertex sequence be $v_1 = v_{i1}v_{i2}\ldots v_{it} = v_t$, which satisfies the following constraint:

$$\max\left|\left\{v_{ij}v_{i(j+1)} \in V(G_1);\ j = 1, 2, \ldots, t-1\right\}\right| \quad (2)$$

here, we give the algorithm for ordering vertices. Let v be a vertex of graph G. We start by defining the set of v-rooted paths, denoted by $P(v)$, then define all the paths starting with the vertex v in G. The set of v-rooted paths $P(v)$ in G can be determined as follows: First, the neighborhood of v is determined; next, the neighborhood of each vertex in $\Gamma(v)$ that is represented as $\Gamma^2(v)$ without v is obtained. Thus, $\Gamma^3(v)$, $\Gamma^4(v)$, etc. are obtained by the same method above, which is called a pruning algorithm. Because the number of vertices in each subgraph is less than 20, this algorithm can be implemented easily.

Suppose that G_1 with $V(G) = \{v_1, v_2, \ldots, v_t\}$ is a subgraph, where v_1 and v_t are the bridge vertices. We now give the algorithm for ordering the vertices of the subgraph G_1:

Step 1: Determine on the rooted paths $P(v_1)$ and $P(v_t)$. If the v_1-rooted paths $P(v_1)$ contain a Hamiltonian path of subgraph G_1, then order the vertices according to the Hamiltonian path. Otherwise, order the vertices according to the following steps.

Step 2: If $P(v_1) \cup P(v_t) = V(G_1)$, then we consider path P_1 in $P(v_1)$ and path P_2 in $P(v_t)$, respectively, where P_1 and P_t satisfy the following conditions: ① $P_1 = v_1v_{i2}\ldots v_{ir}$ and $P_t = v_{is}v_{i(s+1)}\ldots v_t$; ② $\max\{|E(P_1)| + |E(P_t)|\}$; and ③ $V(P_1) \cap V(P_2) = \varnothing$.

Step 3: If $V' = V(G_1) - P(v_1) \cup P(v_t) = \varnothing$, then we further find the longest path from the induced subgraph $G[V']$ and represent it as $P' = u_1u_2\ldots u_m$. If $V'' = V(G_1) - V(P_1) - V(P_t) - V' = \varnothing$, then the ordering of vertices for the subgraph G_1 is $v_1v_{i2}v_{i3}\ldots v_{ir}u_1u_2\ldots u_mv_{is}v_{i(s+1)}\ldots v_t$. Otherwise, if $V'' = V(G_1) - V(P_1) - V(P_t) - V' = \{u_{i1}, u_{i2}, \ldots, u_{iq}\} \neq \varnothing$, then the ordering of vertices for the subgraph G_1 is $v_1v_{i2}v_{i3}\ldots v_{ir}u_1u_2\ldots u_mu_{i1}u_{i2}\ldots u_{iq}v_{is}v_{i(s+1)}\ldots v_t$.

Consider subgraph G'_3 with the bridge vertices 1 and 13 as an example (Fig. 3). The path 1–10–11–2 is picked out from $P(1)$, and the matched path 13–14–7–8–15–3–9–4 is also obtained from

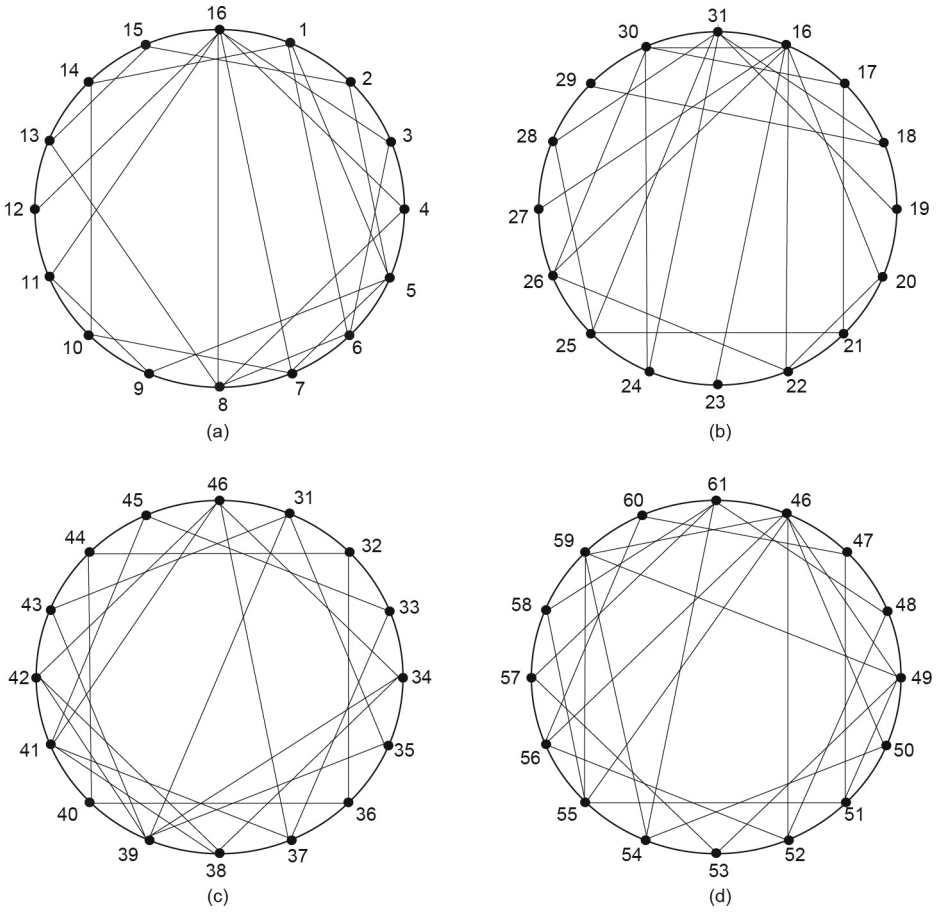

Fig. 4. Four primary subgraphs of graph G (shown in Fig. 1). (a) Subgraph G_1; (b) subgraph G_2; (c) subgraph G_3; (d) subgraph G_4.

$P(13)$. Then the path 5–6 is found in the remaining vertices. Finally, $V'' = \{12\}$. A vertex sequence of G'_3 and the edges incident to the vertices are shown in Fig. 5(a). A similar vertex sequence of G'_3 is shown in Fig. 5(b).

The ordering vertices 1–2–…–16 in G_1 in Fig. 4 are the best order because P_1 = 1–2–…–16. P_1 = 1–2–…–16 is a Hamiltonian path of G_1, and 1 and 16 are the bridge vertices in G_1. Another ordering of the vertices in G_1 is 1–3–5–7–9–11–13–15–2–4–6–8–10–12–14–16, which has only five edges ($\{5,7\}$, $\{9,11\}$, $\{13,15\}$, $\{15,2\}$, and $\{6,8\}$).

Step 4: Relabel the vertices in the subgraphs. Let the vertex sequence be $v_1 v_2 \ldots v_t$ before labeling, and $v_{\sigma(1)} v_{\sigma(2)} \ldots v_{\sigma(t)}$ after reordering. Obviously, the latter vertex ordering is obtained by the permutation of the vertex subscript:

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & t \\ \sigma(1) & \sigma(2) & \cdots & \sigma(t) \end{pmatrix}$$

For example, for the vertex ordering of subgraph G'_3 in Fig. 5, the mapping of the reordered vertices can be shown as follows:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 10 & 11 & 2 & 5 & 6 & 12 & 4 & 9 & 3 & 15 & 8 & 7 & 14 & 13 \end{pmatrix}$$

In general, the vertices of a subgraph should be relabeled after ordering, and the vertex $\sigma(i)$ will be relabeled by i, where i = 1, 2, …, t.

The chromatic number k in this article is 3, and the color set $C(3)$ = {red, yellow, blue} = {r, y, b}. The variables r_i, y_i, and b_i indicate that the color of v_i can be chosen as red, yellow, or blue.

Considering subgraph G_1, where $V(G_1) = \{v_1, v_2, \ldots, v_t\}$, the vertices v_1 and v_t are the bridge vertices in G_1. In our model, the bridge vertices are always adjacent, so vertices v_1 and v_t can be colored red (r_1) and blue (b_t) in advance. (It should be noted that vertex v_t can be colored red or yellow, of course, when it is not adjacent with a vertex colored blue. The true solutions with r_t or y_t can be obtained by color permutation according to the coloring with b_t.) The vertices belonging to the set $\Gamma(v_1)$ will be blue or yellow, and those belonging to $\Gamma(v_t)$ will be red or yellow. The color sets of the vertices of the four subgraphs in Fig. 4 are shown in Fig. 6.

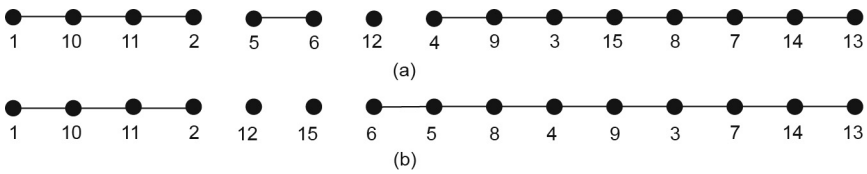

Fig. 5. The ordering vertices and set of the corresponding 11 edges for subgraph G'_3 in Fig. 3(d).

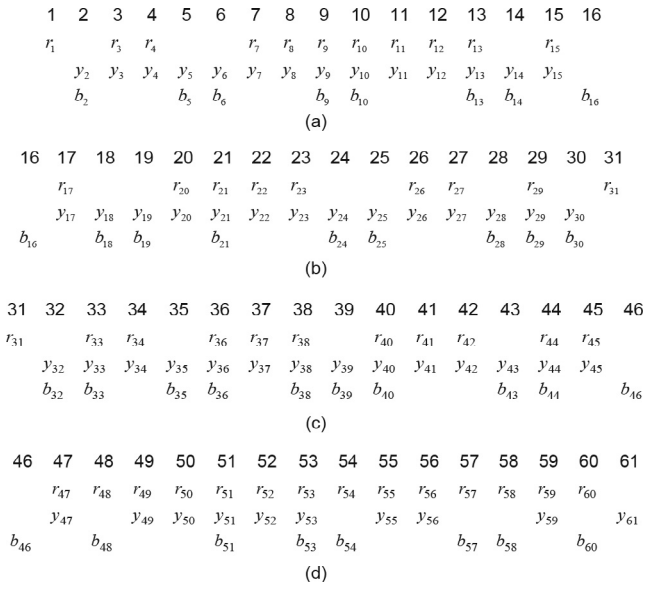

Fig. 6. All possible colors for the vertices of the subgraph when $C(1) = \{r_1\}$, $C(16) = \{b_{16}\}$, $C(31) = \{r_{31}\}$, $C(46) = \{b_{46}\}$, and $C(61) = \{y_{61}\}$ are designated in advance. Color set of each vertex in (a) subgraph G_1, (b) subgraph G_2, (c) subgraph G_3, and (d) subgraph G_4.

This method can be generalized to any situation where $k > 3$. If a graph G can be colored by k ($k \geq \chi(G)$) colors, then the color set is denoted by $C_k(G)$, which can be defined as $C_3(G) = \{c_1, c_2, \ldots, c_k\}$. Then the k-coloring problem of a given graph G can be solved if we find a map f: $V(G) \rightarrow C_3(G)$ with the property that $\forall u v \in E(G)$, $f(u) \neq f(v)$. The other steps are completely the same as for the situation where $k = 3$.

This method greatly reduces the complexity of the initial solution space. The relevant formula and the analysis are given in Sections 3.2 and 5.2, respectively.

3.4. Encoding

As DNA molecules work through specific hybridization in the computing process, encoding in DNA computing will be constrained by many factors such as the Gibbs free energy of chemistry, melting temperature, similarity of DNA sequences, length of DNA sequences, experimental environment, scale of problems, and so forth. Therefore, encoding is a very complicated and difficult problem in DNA computing, and is considered to be an NP-complete problem by several scholars. Many algorithms of DNA encoding based on various constraints have been formulated thus far. To provide good DNA strands for both specific hybridization and PCR in this article, the DNA strands are designed based on the following constraints:

- All sequences have no occurrence of five or more consecutive identical bases;
- The GC content of all sequences is 40%–60%;
- There are no more than eight of the same bases between any two sequences;
- There are no more than four matched bases in every sequence;
- For every sequence, five consecutive bases in both the 3′ and 5′ ends are mismatched with any five bases in other sequences;
- The absolute value of the change in the Gibbs free energy of chemistry ($|\Delta G|$) of the primer dimer formed by its own sequence is no more than 6.0 kcal·mol^{-1} (1 cal = 4.1868 J); and
- The absolute value of the change in the Gibbs free energy of chemistry ($|\Delta G|$) of the primer dimer is no more than 9.0 kcal·mol^{-1}, and the absolute value of the change in the Gibbs free energy of chemistry ($|\Delta G|$) of the primer dimer formed at the 3′ end is no more than 6.0 kcal·mol^{-1}.

Based on these constraints, the method described in Ref. [27] was used to design the DNA strands; next, the proper sequence was selected using the software Primer Premier 5.0.

It is easy to prove the following theorem.

Theorem 3.1: Let G be a 3-coloring graph with bridge vertices v_1 and v_n. According to the possible color set of each vertex, it can be determined that the number of DNA sequences representing vertex colors for any 3-coloring graph with n vertices is

$$N_{\mathrm{DNA}} \leqslant 3n - d(v_1) - d(v_n) - 2(3-1) \tag{3}$$

In this article, 129 oligonucleotides (Table 1) were designed to represent the possible colors of the vertices in graph G. These oligonucleotides and their Watson–Crick complementary sequence were denoted by x_i and $\overline{x_i}$, where $i = 1, 2, \ldots, 61$, respectively.

3.5. Determining probes

The advantages of DNA computing are its huge parallelism and its great capacity for information storage. Over decades of research, it has been recognized that the biological activity in artificial DNA molecules is the main feature of the DNA computer. In particular, the gravitation between hydrogen bonds dominates the process. By making use of this feature, most of the false solutions can be eliminated from the initial solution space. This eliminating process cannot be carried out by electronic computers. To implement this idea, the method of designing hybridization probes plays a significant role in deleting false solutions.

In this model, the method of designing probes is described as follows (with chromatic number $k = 3$ as an example):

Step 1: Determine the color sets of each vertex. Let $V(G) = \{v_1, v_2, \ldots, v_n\}$. Without loss of generality, let its ordering of vertices be $v_1 v_2 \ldots v_n$. The set of possible colors for each vertex v_i is denoted by C_{v_i}, where $i = 1, 2, \ldots, n$.

Step 2: Determine the set of probes. The probes are used to constitute the initial solution space. Probes are only set up between two consecutive labeled vertices v_i and v_{i+1}, denoted by $\overline{x_i x_{i+1}}$, where $x_i \in C_{v_i}$, $x_{i+1} \in C_{v_{i+1}}$, and $i = 1, 2, \ldots, n-1$. If $\exists e = v_i v_{i+1} \in E(G)$, then x_i and x_{i+1} cannot have the same color, and vice versa. For example, $C_{v_7} = \{r_7, y_7\}$, $C_{v_8} = \{r_8, y_8\}$, and $e = v_7 v_8 \in E(G_1)$ exist in subgraph G_1 (see Fig. 4), so only two probes ($\overline{r_7 y_8}$, $\overline{y_7 r_8}$) are constructed.

Step 3: Determine the DNA sequence for each probe. The probes $\overline{x_i x_{i+1}}$ consist of the last half of the DNA sequence of x_i and the first half of the DNA sequence of x_{i+1}. In this article, the length of the probe is 20 bases. For example, if $y_7 = 5'$-AATACGCACTC ATCACATCG-3′ and $r_8 = 5'$-GACCTTACCGTTTAGAGTCG-3′, then $\overline{y_7 r_8} = 5'$-CGGTAAGGTCCGATGTGATG – 3′.

To explain probe sets clearly, we utilize the graph theory method. This method is very useful for reducing initial solution space and theoretical analysis.

The probe graph of a graph G with n vertices, denoted by $B(G)$ (abbreviated as B), is defined as follows:

$$\begin{aligned} V(B) &= \bigcup_{i=1}^{n} C_{v_i}, \quad \text{where } E(B) \\ &= \bigcup_{i=1}^{n} \{\overline{xz}; \overline{xz} \text{ is a probe}, x \in C_{v_i}, z \in C_{v_{i+1}}\} \end{aligned} \tag{4}$$

According to the definition of probes and Fig. 6, the 3-coloring probe graphs of four subgraphs (see Fig. 4) are shown as B_1, B_2, B_3, and B_4 in Fig. 7. It is easy to identify information on probe numbers and distribution from each probe graph.

Table 1
DNA sequences for each x_i.

x_i	DNA sequence	x_i	DNA sequence
r_1	5′-CTGGT CCTCT CCTCT AATCC-3′	y_2	5′-AAGAG AGAAC CGAAC TGTCC-3′
b_2	5′-ACTTG AGCAC TGACC TGACA-3′	r_3	5′-AAGAG GCTAC GGACA CTACT-3′
y_3	5′-AAGGA TGAAC CATCG CACAG-3′	r_4	5′-TAGGT GCTAC AGATT CGTCC-3′
y_4	5′-AAGTC TGAAC GCCTA CTCAC-3′	y_5	5′-CAGAA CACAG GTATG CGATT-3′
b_5	5′-AAGAC CACAC CACAG CATTC-3′	y_6	5′-CGTGA TTGTT GGACT ATTGG-3′
b_6	5′-CCTTG TAGAC CCAGA TGTTC-3′	r_7	5′-CGTTG CTCTG AATAG TTGCC-3′
y_7	5′-AATAC GCACT CATCA CATCG-3′	r_8	5′-GACCT TACCG TTTAG AGTCG-3′
y_8	5′-AATAC ATCAG AGCGG AGACC-3′	r_9	5′-ATGGT GGAAA TCTAC TCGCC-3′
y_9	5′-AAGGC TACAA ACTCA CCGAC-3′	b_9	5′-AGGAG GTTTG TTAGC CAGTC-3′
r_{10}	5′-ACAGA AAGAA ACTCG CTTCG-3′	y_{10}	5′-GAAGA TGAAC CAGCC TAACC-3′
b_{10}	5′-AAGTG AACAG TGTGA CCACC-3′	r_{11}	5′-TCACA TTAGT GTCAC AGCGG-3′
y_{11}	5′-CAGAG ACAAG ACGAA CCTGT-3′	r_{12}	5′-TAGAA GAAGC AACCG TCTGT-3′
y_{12}	5′-AACTT GTTCC ACACA CCCTC-3′	r_{13}	5′-GCTTA TGTAT CCTGG CACTG-3′
y_{13}	5′-ATGAG TTACA AGCAC CACGC-3′	b_{13}	5′-TACAG GGTCT TCAGA ACGAT-3′
y_{14}	5′-ATGTC TCGTC AGGAT GTCGT-3′	b_{14}	5′-TTCCC TACTA CCTTC CCAAG-3′
r_{15}	5′-ATGCC TCAAC AACTC CTGCT-3′	y_{15}	5′-ATGGT ATGAA GCCTG ACTCG-3′
b_{16}	5′-GGATT GTATT GGCGA TGATG-3′	r_{17}	5′-TACAT TCAAG GACGA CAGGT-3′
y_{17}	5′-CAAAG TGTAG GCAGG GTAAC-3′	y_{18}	5′-AAGCG GTAGA CACGA TTCAC-3′
b_{18}	5′-TAGAG TCCAC CGAAG ATAGC-3′	y_{19}	5′-ACTGC TAATG ACTCG TTCCG-3′
b_{19}	5′-AGTAT CTGTC CTGTC TCACC-3′	r_{20}	5′-GTGGT CGTAG ATGTC ACTCC-3′
y_{20}	5′-AACTA ACGAC CAGAG CCGAT-3′	r_{21}	5′-TAGTC ATAAG TGACC TCGGC-3′
y_{21}	5′-ATCTC CATCC AACCA TCCAG-3′	b_{21}	5′-AATCA ACTGG TCACG ACTGC-3′
r_{22}	5′-GTTAT GAGTC GCAGC ACACG-3′	y_{22}	5′-TAGTG CGGAA CCTAT CTTGC-3′
r_{23}	5′-AAGTG GAGAC ACTCA CTACC-3′	y_{23}	5′-AAGTA TCAGA CAGCC ATCCG-3′
y_{24}	5′-AAGTT GAAGG CTTAC GAGAC-3′	b_{24}	5′-ATGTC AAGCA TACCG TCACC-3′
y_{25}	5′-ATGTG TATGT TGCGA CAAGC-3′	b_{25}	5′-GATTA TCGTC CAGCC TTCTC-3′
r_{26}	5′-CCTCC GTAGT TATTG ATGCC-3′	y_{26}	5′-GTTAC GGTTG ACTCT GCTGA-3′
r_{27}	5′-CACTT CTACC CTCAA CCTCA-3′	y_{27}	5′-TAGTA GAAAG CCGAC CACTC-3′
y_{28}	5′-CGTGG AAGTC ACTAA GGTCT-3′	b_{28}	5′-ATTCT TCACT GAGGT GCTGG-3′
r_{29}	5′-TAAGT GAGAA TGCCA GTTGC-3′	y_{29}	5′-CGAGA TGTTG TAAAG GCTGC-3′
b_{29}	5′-GGTAT GTAAC AAGAC GCACG-3′	y_{30}	5′-GGTCA TTATG GGCAT AGTGG-3′
b_{30}	5′-AACAT TTACT CGTCG TTCGC-3′	r_{31}	5′-GCTCA GGAGT GTTTA TGACC-3′
y_{32}	5′-TAGTC CATCG GCAAG GTTCT-3′	b_{32}	5′-AACAC CACAG AGCAT TCACG-3′
r_{33}	5′-AAGTG CTGAA ACCGT GGAGT-3′	y_{33}	5′-TACTG AGAAC GCTCG CTCTT-3′
b_{33}	5′-ATGTC ATAAT GCCGT TCCTG-3′	r_{34}	5′-GATGT TGCCA TAACT GCCTG-3′
y_{34}	5′-CTACT TATTC GTCAG CGTCG-3′	y_{35}	5′-GATGA TTACA CTCGC ACAGG-3′
b_{35}	5′-GAACA TAATG GACCG ACCTC-3′	r_{36}	5′-ATCTG TGCTA TCTCG TGCTC-3′
y_{36}	5′-AACTT ACCAT TGGCT TCTGC-3′	b_{36}	5′-GATAA ACGAG TTCGC ATAGC-3′
r_{37}	5′-GCAGT AGACG ATACG ACTCC-3′	y_{37}	5′-CGAAG AAGAT TACCC AGAGG-3′
r_{38}	5′-AAGCA CTCAA CAGTA CGAGC-3′	y_{38}	5′-GGTGA TGGTT GAAAG TCTCC-3′
b_{38}	5′-TAGAG ATTGG ACGGA AGACG-3′	y_{39}	5′-TAGGT ATAGG TCGTT GAGCC-3′
b_{39}	5′-CAAGT CACAA TCGTA GGTGC-3′	r_{40}	5′-GCTAA CAGTG GTCAG ACACG-3′
y_{40}	5′-AATAC CACCT GACTG CGTAG-3′	b_{40}	5′-AATAC ACTAT CCAGC GACGG-3′
r_{41}	5′-GGTGT AAGCC TCCGT ATTAG-3′	y_{41}	5′-GGAAC CTACT CTGGA TGAAG-3′
r_{42}	5′-GGTAA CGATC CTGAT AACGC-3′	y_{42}	5′-GCTGT CCAAC CAGGT CTTAC-3′
y_{43}	5′-CCTAC ACATC AATCA GCACC-3′	b_{43}	5′-CCTTA CAAAT CGCCT ATGGT-3′
r_{44}	5′-CCAAA CTTGC TTACT TCAGG-3′	y_{44}	5′-CAAAG AGTTA GTCGG GTCTG-3′
b_{44}	5′-CTTCT ATGTT TAGCC CGAGG-3′	r_{45}	5′-GCAGG ACAAG GCTCA TAGTT-3′
y_{45}	5′-GTGAC GCCAT CATTT GAGAT-3′	b_{46}	5′-CTATC AGAAA CCCGT CAGAG-3′
r_{47}	5′-GCTGA CTTCA CGGAT TTGGA-3′	y_{47}	5′-CGAAG GACTT AGTAA CGAGG-3′
r_{48}	5′-CCTAT GCCTA AATGG TGTCG-3′	b_{48}	5′-CCTGT CCGAT AGAAT AGTGC-3′
r_{49}	5′-AGTTG CGTCC ACGAA AGTAG-3′	y_{49}	5′-GCACT CCCAA TGTGT TATGA-3′
r_{50}	5′-AGGCT CCATC TTGAG AACTG-3′	y_{50}	5′-GCTGG CGACT ACTAT TTACG-3′
r_{51}	5′-GCTCT CCCTT ATGGA ATGAT-3′	y_{51}	5′-CACTA AACAA CGCAG GGTTC-3′
b_{51}	5′-CAGCA ACCAC ATCGG TGATA-3′	r_{52}	5′-GACCT CCTGA AAGAG TACGA-3′
y_{52}	5′-GTCAC CTGCT AGGAG GATTC-3′	r_{53}	5′-GAGTC GTCGG AGATA AGGTT-3′
y_{53}	5′-GAATA CCTGT GCTAC CGAGT-3′	b_{53}	5′-GGATA GCGAT TGACT GAACG-3′
r_{54}	5′-CTGAG TCCTT TGAGT AAGCC-3′	b_{54}	5′-CAGAT AGACT CCGCT GAGGT-3′
r_{55}	5′-GAGTT CCATT GTGGC AGAAG-3′	y_{55}	5′-GCATT TCACA GTCTT CTCGC-3′
r_{56}	5′-GATTA CTCCA CCCTC GTGTA-3′	y_{56}	5′-CAGTT ACATT GAGCG GAAGC-3′
r_{57}	5′-CAAGT ATGGC TCACA TTCGT-3′	b_{57}	5′-CAAAC AGGCG TCTCT TTATG-3′
r_{58}	5′-CAAGC AGCAC GATGA CTCTA-3′	b_{58}	5′-GACTT GCTCT GCGTG AGATT-3′
r_{59}	5′-GACAT TGCTG AATCA GTGGT-3′	y_{59}	5′-GCTAC TGCTA AGGGT AATGC-3′
r_{60}	5′-GCACT GTATG ACAGG TCACG-3′	b_{60}	5′-CGTTA GGACC TGGGA TAATC-3′
y_{61}	5′-GATTA CTCCA CCCTC GTGTA-3′		

3.6. Construction of the initial solution space

The method presented by Adleman [2] is used to construct the initial solution space of each deduced subgraph. The DNA sequences representing the possible solutions in the initial solution space are amplified in a series of operations:

- First, the 5′-ends of DNA sequences x_i are phosphorylated by T4 polynucleotide kinase (PNK);
- Second, the product of phosphorylation and probes are mixed in 10 × T4 PNK buffer for annealing;
- Third, the annealed products are ligated by T4 DNA ligase in 10 × T4 DNA ligase buffer; and

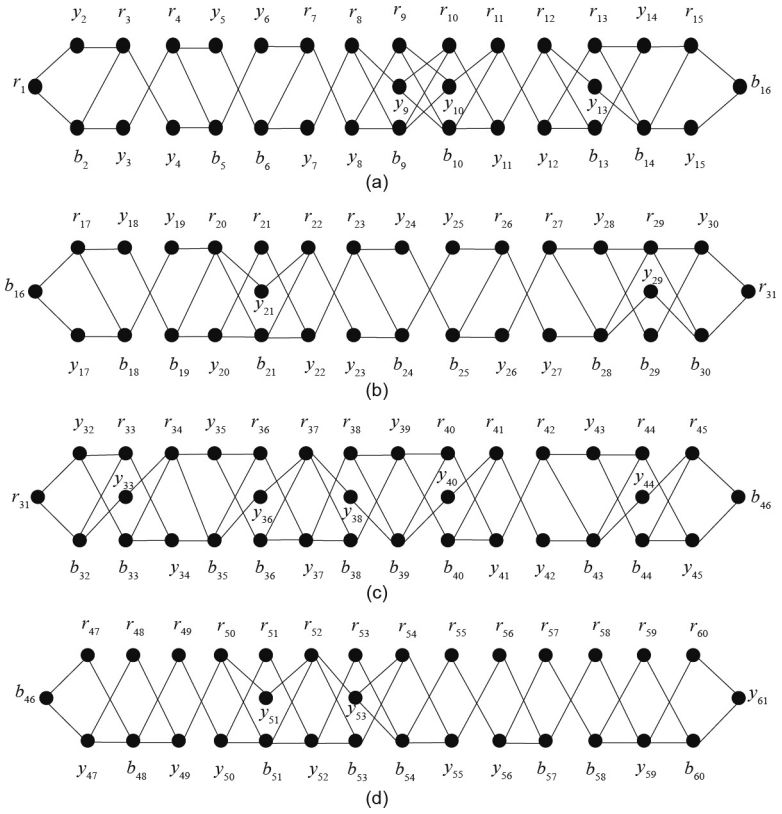

Fig. 7. Probe graphs for the subgraph shown in Fig. 4. (a) B_1; (b) B_2; (c) B_3; (d) B_4.

- Finally, the ligated products are amplified by PCR with the primer pair $\langle r_1, \overline{x}_t \rangle$. The amplified DNA strands represent all possible solutions of each subgraph.

In this article, the initial solution space of subgraph G_j is denoted by $S(G_j)$, where $j = 1, 2, \ldots, m$, and m is the number of subgraphs. Let $t = |G_j|$ and $V(G) = \{v_1, v_2, \ldots, v_t\}$. Then $S(G_j) = \{x_1 x_2 \ldots x_t\}$, where $x_i \in C_{v_j} \subset \{r, b, y\}$, and $i = 1, 2, \ldots, t$. If each x_i has l bases, then the length of the DNA sequence in the initial solution space will be $t \times l$.

It should be noted that if $e = v_i v_{i+1} \in E(G)$, then x_i and x_{i+1} cannot share the same color when constructing the probes. In this way, the false solutions in which adjacent vertices have the same color will be deleted when constructing the initial solution space. Thus, the more edges $e = \{v_i, v_{i+1}\}$ there are, the more false solutions will be deleted in the initial solution space. The details of the theoretical analysis will be shown in Section 5.2.

3.7. Deleting false solutions

In this section, the methods of deleting the false solutions of each subgraph are described. Let $E_j^1 = \{v_i v_{i+1} \in E(G),\ v_i v_t \in E(G_j);\ i = 1, 2, \ldots, t-1\}$, and $E_j^2 = \{v_1 v_i \in E(G_j),\ i = 2, 3, \ldots, t\}$. Then $E_j^3 = E(G_j) - E_j^1 - E_j^2$ and $j = 1, 2, \ldots, m$.

In fact, the process of deleting false solutions involves eliminating DNA strands that represent the improper coloring of the vertices incident to the edges in E_j^3 from the initial solution space $S(G_j)$. In the process of bio-operation, PCR is used to delete false solutions. To improve the processing speed, the edges in E_j^3 are divided into two groups and each group is operated using different procedures.

(1) Forward Path. In our experiment, a false solution is deleted by a so-called parallel PCR operation on the forward paths. Here, we introduce a definition of forward path.

Let G_1 be a subgraph with $V(G_1) = \{v_1, v_2, \ldots, v_t\}$, and let $v_1 v_2 \ldots v_t$ be a vertex ordering of G_1 according to the method shown in Section 3.3. For $\forall e = v_i v_j \in E(G_1)$, if $i < j$, then $e = v_i v_j$ is a forward edge; otherwise, it is a backward edge. If all the edges in a path are forward edges, then the path is a forward path; otherwise, it is a backward path.

Now, for example, we take a path $P = v_i v_j v_y v_z$, $1 < i < j < y < z < t$ (other paths are treated in the same way). In this path, there are three edges: $v_i v_j$, $v_j v_y$, and $v_y v_z$. First, PCR is done with primer pairs: $\langle x_1, \overline{x_i} \rangle$, $\langle x_i, \overline{x_j} \rangle$, $\langle x_j, \overline{x_y} \rangle$, $\langle x_y, \overline{x_z} \rangle$, and $\langle x_z, \overline{x_t} \rangle$, where x_i and x_j, x_j and x_y, and x_y and x_z should not have the same color at the same time. In this way, the DNA strands are separated into five fragments. Subsequently, these small fragments will be combined gradually to generate DNA strands with $t \times l$ base pairs (bp). Thus, at the end of PCR operations, the DNA strands representing the improper coloring of these vertices incident to the three edges will be eliminated from $S(G_j)$. Detailed experimental methods and the theoretical analysis are given in Sections 4.4 and 5.3, respectively.

(2) Single edge. Some single edges exist in E_j^3 after deleting all false solutions from forward paths. Let $e = v_i v_j$, $x_i \in C_{v_i}$, and $x_j \in C_{v_j}$. PCR is run with primer pairs $\langle x_1, \overline{x_i} \rangle$, $\langle x_i, \overline{x_j} \rangle$, and $\langle x_j, \overline{x_t} \rangle$. Here, x_i and x_j should not have the same color. In this way, the DNA strands representing improper coloring of the vertices incident to the edge are not amplified. That is, the DNA strands representing false solutions are eliminated. When the DNA strands representing improper coloring of the vertices incident to the edges in E_j^3 are all deleted, the amplified DNA strands in $S(G_j)$ represent the colorings of subgraph G_j, where $j = 1, 2, \ldots, m$.

3.8. Subgraph combination and deletion of false solutions

After obtaining the true solutions of each subgraph, these subgraphs are combined to generate secondary subgraphs. Then the process of deleting false solutions continues. This process involves the following three steps:

Step 1: Combine subgraphs G_j and G_{j+1}, where $j = 1, 2, \ldots, m-1$. The DNA strands representing the improper colors of the subgraphs will be linked together to generate new DNA strands by the DNA sequence of the bridge vertices. These new DNA strands will be processed as the initial solution space of the secondary graph $G[V(G_j) \cup V(G_{j+1})]$. If $|V(G_j)| = t_1$ and $|V(G_{j+1})| = t_2$, then the length of the DNA strands generated is $(t_1 + t_2 - 1) \times l$.

Step 2: Delete the false solutions of secondary graph $G[V(G_j) \cup V(G_{j+1})]$ according to the method described in Section 3.7. That is, the DNA strands representing the color of the vertices incident to the edge in $E_{jj+1} = \{uv \in E(G); u \in V(G_j), v \in V(G_{j+1})\}$ will be amplified, but the false solutions will be eliminated.

Step 3: Repeat Steps 1 and 2 for the secondary and new combined subgraphs until the original graph is completely reshaped. Thus, the DNA strands representing the true solutions of G will be obtained.

3.9. Detecting the true solutions

The DNA strands representing the true solutions are sequenced and analyzed by Gene Tools to obtain the proper coloring of the graph.

4. Implementation of the model

To implement this DNA computing model, consider the graph with 61 vertices shown in Fig. 1 as an example. The scale of this problem is the largest to be solved by DNA computing. The protocol of the assay is given in this section. The result illustrates that the computational capability of the DNA computing model proposed here is as high as $O(3^{59})$.

4.1. Dividing the graph and determining the color set

According to the methods demonstrated in Section 3.2, graph G (Fig. 1) can be divided into four subgraphs (Fig. 4); the color set of each vertex is shown in Fig. 6.

4.2. Encoding

According to the methods in Section 3.5, 129 oligonucleotides (Table 1) were designed to represent the possible colors of all vertices, and 185 probes were constructed according to the probe graph (Fig. 7).

These probes are: $\overline{r_1y_2}$, $\overline{r_1b_2}$; $\overline{y_2r_3}$, $\overline{b_2r_3}$, $\overline{b_2y_3}$; $\overline{r_3y_4}$, $\overline{y_3r_4}$; $\overline{r_4y_5}$, $\overline{r_4b_5}$, $\overline{y_4b_5}$; $\overline{y_5b_6}$, $\overline{b_5y_6}$; $\overline{y_6r_7}$, $\overline{b_6r_7}$, $\overline{b_6y_7}$; $\overline{r_7y_8}$, $\overline{y_7r_8}$; $\overline{r_8y_9}$, $\overline{r_8b_9}$, $\overline{y_8r_9}$, $\overline{y_8b_9}$; $\overline{r_9y_{10}}$, $\overline{r_9b_{10}}$, $\overline{y_9r_{10}}$, $\overline{y_9b_{10}}$, $\overline{b_9r_{10}}$, $\overline{b_9y_{10}}$; $\overline{r_{10}y_{11}}$, $\overline{y_{10}r_{11}}$, $\overline{b_{10}r_{11}}$, $\overline{b_{10}y_{11}}$; $\overline{r_{11}y_{12}}$, $\overline{y_{11}r_{12}}$; $\overline{r_{12}y_{13}}$, $\overline{r_{12}b_{13}}$, $\overline{y_{12}r_{13}}$, $\overline{y_{12}b_{13}}$; $\overline{r_{13}y_{14}}$, $\overline{r_{13}b_{14}}$, $\overline{y_{13}b_{14}}$, $\overline{b_{13}y_{14}}$; $\overline{y_{14}r_{15}}$, $\overline{b_{14}r_{15}}$, $\overline{b_{14}y_{15}}$; $\overline{r_{15}b_{16}}$, $\overline{y_{15}b_{16}}$; $\overline{b_{16}r_{17}}$, $\overline{b_{16}y_{17}}$; $\overline{r_{17}y_{18}}$, $\overline{r_{17}b_{18}}$, $\overline{y_{17}b_{18}}$; $\overline{y_{18}b_{19}}$, $\overline{b_{18}y_{19}}$; $\overline{y_{19}r_{20}}$, $\overline{b_{19}r_{20}}$, $\overline{b_{19}y_{20}}$; $\overline{r_{20}y_{21}}$, $\overline{r_{20}b_{21}}$, $\overline{y_{20}r_{21}}$, $\overline{y_{20}b_{21}}$; $\overline{r_{21}y_{22}}$, $\overline{y_{21}r_{22}}$, $\overline{b_{21}r_{22}}$, $\overline{b_{21}y_{22}}$; $\overline{r_{22}y_{23}}$, $\overline{y_{22}r_{23}}$; $\overline{r_{23}y_{24}}$, $\overline{r_{23}b_{24}}$, $\overline{y_{23}b_{24}}$; $\overline{y_{24}b_{25}}$, $\overline{b_{24}y_{25}}$; $\overline{y_{25}r_{26}}$, $\overline{b_{25}r_{26}}$, $\overline{b_{25}y_{26}}$; $\overline{r_{26}y_{27}}$, $\overline{y_{26}r_{27}}$; $\overline{r_{27}y_{28}}$, $\overline{r_{27}b_{28}}$, $\overline{y_{27}b_{28}}$; $\overline{y_{28}r_{29}}$, $\overline{y_{28}b_{29}}$, $\overline{b_{28}r_{29}}$, $\overline{b_{28}y_{29}}$; $\overline{r_{29}y_{30}}$, $\overline{r_{29}b_{30}}$, $\overline{y_{29}b_{30}}$, $\overline{b_{29}y_{30}}$; $\overline{y_{30}r_{31}}$, $\overline{b_{30}r_{31}}$; $\overline{r_{31}y_{32}}$, $\overline{r_{31}b_{32}}$; $\overline{y_{32}r_{33}}$, $\overline{y_{32}b_{33}}$, $\overline{b_{32}r_{33}}$, $\overline{b_{32}y_{33}}$; $\overline{r_{33}y_{34}}$, $\overline{y_{33}r_{34}}$, $\overline{b_{33}r_{34}}$, $\overline{b_{33}y_{34}}$; $\overline{r_{34}y_{35}}$, $\overline{r_{34}b_{35}}$, $\overline{y_{34}b_{35}}$; $\overline{y_{35}r_{36}}$, $\overline{y_{35}b_{36}}$, $\overline{b_{35}r_{36}}$, $\overline{b_{35}y_{36}}$; $\overline{r_{36}y_{37}}$, $\overline{y_{36}r_{37}}$, $\overline{b_{36}r_{37}}$, $\overline{b_{36}y_{37}}$; $\overline{r_{37}y_{38}}$, $\overline{r_{37}b_{38}}$, $\overline{y_{37}r_{38}}$, $\overline{y_{37}b_{38}}$; $\overline{r_{38}y_{39}}$, $\overline{r_{38}b_{39}}$, $\overline{y_{38}b_{39}}$, $\overline{b_{38}y_{39}}$; $\overline{y_{39}r_{40}}$, $\overline{y_{39}b_{40}}$, $\overline{b_{39}r_{40}}$, $\overline{b_{39}y_{40}}$; $\overline{r_{40}y_{41}}$, $\overline{y_{40}r_{41}}$, $\overline{b_{40}r_{41}}$, $\overline{b_{40}y_{41}}$; $\overline{r_{41}y_{42}}$, $\overline{y_{41}r_{42}}$; $\overline{r_{42}y_{43}}$, $\overline{r_{42}b_{43}}$, $\overline{y_{42}b_{43}}$; $\overline{y_{43}r_{44}}$, $\overline{y_{43}b_{44}}$, $\overline{b_{43}r_{44}}$, $\overline{b_{43}y_{44}}$; $\overline{r_{44}y_{45}}$, $\overline{y_{44}r_{45}}$, $\overline{b_{44}r_{45}}$, $\overline{b_{44}y_{45}}$; $\overline{r_{45}b_{46}}$, $\overline{y_{45}b_{46}}$; $\overline{b_{46}r_{47}}$, $\overline{b_{46}y_{47}}$; $\overline{r_{47}b_{48}}$, $\overline{y_{47}r_{48}}$, $\overline{y_{47}b_{48}}$; $\overline{r_{48}y_{49}}$, $\overline{b_{48}r_{49}}$, $\overline{b_{48}y_{49}}$; $\overline{r_{49}y_{50}}$, $\overline{y_{49}r_{50}}$; $\overline{r_{50}y_{51}}$, $\overline{r_{50}b_{51}}$, $\overline{y_{50}r_{51}}$, $\overline{y_{50}b_{51}}$; $\overline{r_{51}y_{52}}$, $\overline{y_{51}r_{52}}$, $\overline{b_{51}r_{52}}$, $\overline{b_{51}y_{52}}$; $\overline{r_{52}y_{53}}$, $\overline{r_{52}b_{53}}$, $\overline{y_{52}r_{53}}$, $\overline{y_{52}b_{53}}$; $\overline{r_{53}b_{54}}$, $\overline{y_{53}r_{54}}$, $\overline{y_{53}b_{54}}$, $\overline{b_{53}r_{54}}$; $\overline{r_{54}y_{55}}$, $\overline{b_{54}r_{55}}$, $\overline{b_{54}y_{55}}$; $\overline{r_{55}y_{56}}$, $\overline{y_{55}r_{56}}$; $\overline{r_{56}b_{57}}$, $\overline{y_{56}r_{57}}$, $\overline{y_{56}b_{57}}$; $\overline{r_{57}b_{58}}$, $\overline{b_{57}r_{58}}$; $\overline{r_{58}y_{59}}$, $\overline{b_{58}r_{59}}$, $\overline{b_{58}y_{59}}$; $\overline{r_{59}b_{60}}$, $\overline{y_{59}r_{60}}$, $\overline{y_{59}b_{60}}$; and $\overline{r_{60}y_{61}}$, $\overline{b_{60}y_{61}}$.

All the oligonucleotides used in this article were synthesized by Sangon Biotech (Shanghai) Co., Ltd.

4.3. Implementing the initial solution space of each subgraph

Here, we use the subgraph G_1 as an example to show how to construct the initial solution space; the result is shown in Fig. 8(a). The steps and reaction systems were as follows:

Step 1: Phosphorylation. DNA sequences x_i, where $v_i \in V(G_1)$, $x_i \in C_v$, and $i = 1, 2, \ldots, 16$, were phosphorylated at the 5′ end by T4 PNK. In a sterile 0.2 mL amplification tube, the phosphonated reaction components were: 16.5 μL of oligonucleotide (0.5 μL for each x_i; $v_i \in V(G_1)$, $x_i \in C_v$, $i = 1, 2, \ldots, 16$), 3 μL of T4 PNK, 3 μL of buffer forT4 PNK, 5 μL of adenosine triphosphate (ATP) (10 mmol·L^{-1}), and 2.5 μL of double-distilled water. The total volume of the mixture was 30 μL. The mixture was incubated at 37 °C for an hour.

Step 2: Annealing. In a sterile 0.2 mL amplification tube, the annealing reaction components were: 30 μL of the phosphorylated product, 23 μL of probes (0.5 μL for each probe), 3 μL of buffer for T4 PNK, and 4 μL of double-distilled water for a total volume of 60 μL. The mixture was incubated at 94 °C for 5 min and at 50 °C for 10 min, and was then slowly cooled to room temperature.

Step 3: Ligation. In a sterile 0.2 mL amplification tube, the ligation reaction components were: 6 μL of the annealing product, 2 μL of T4 DNA ligase, and 2 μL of buffer for T4 DNA ligase. The total volume of the mixture was 10 μL. The mixture was incubated at 16 °C overnight.

Step 4: PCR amplification. The ligation product was amplified by the primer pairs $\langle r_1, \overline{b_{16}} \rangle$. In a sterile 0.2 mL amplification tube, the PCR reaction components were: 1 μL of the template, 2 μL of the primer pairs (1 μL of each), 4 μL of dNTP Mix (2.5 mmol·L^{-1} for each), 0.5 μL of the *Taq* polymerase, and 5 μL of the buffer for the *Taq* polymerase, in a total volume of 30 μL. The reaction conditions were: 94 °C for 5 min; then 94 °C for 30 s, 54 °C for 30 s, and 72 °C for 45 s for 35 cycles; then 72 °C for 10 min.

After PCR, 4% agarose gel electrophoresis was used to display the products, and the 320 bp bands were cut and purified with a Gel Extraction Kit (U-gene) in a final volume of 50 μL. The DNA strands representing the possible solutions of subgraph G_1 were denoted by $S(G_1)$, where $S(G_1) = \{x_1x_2 \ldots x_{16}\}$.

These steps were also used to construct the initial solution space of G_2, G_3, and G_4; the results are shown in Fig. 8(b–d). These results were denoted by $S(G_2)$, $S(G_3)$, and $S(G_4)$, respectively, where $S(G_2) = \{x_{16}x_{17} \ldots x_{31}\}$, $S(G_3) = \{x_{31}x_{32} \ldots x_{46}\}$, and $S(G_4) = \{x_{46}x_{47} \ldots x_{61}\}$, with $x_i \in C_{v_i}$ and $v_i \in V(G)$.

PCR was also used to test the completeness of each initial solution space of each subgraph. Taking $S(G_1)$ as an example, a 50-fold dilution of the initial solution space $S(G_1)$ was amplified with primer pairs: $\langle x_1, \overline{x_i} \rangle$, where $i = 1, 2, \ldots, 16$. These products are shown in Fig. 8(e,f).

4.4. Deleting false solutions of the subgraph

In this section, we describe how to delete false solutions of the subgraph. PCR is also used to delete false solutions. According to the different walks, PCR operations were classified into three basic types:

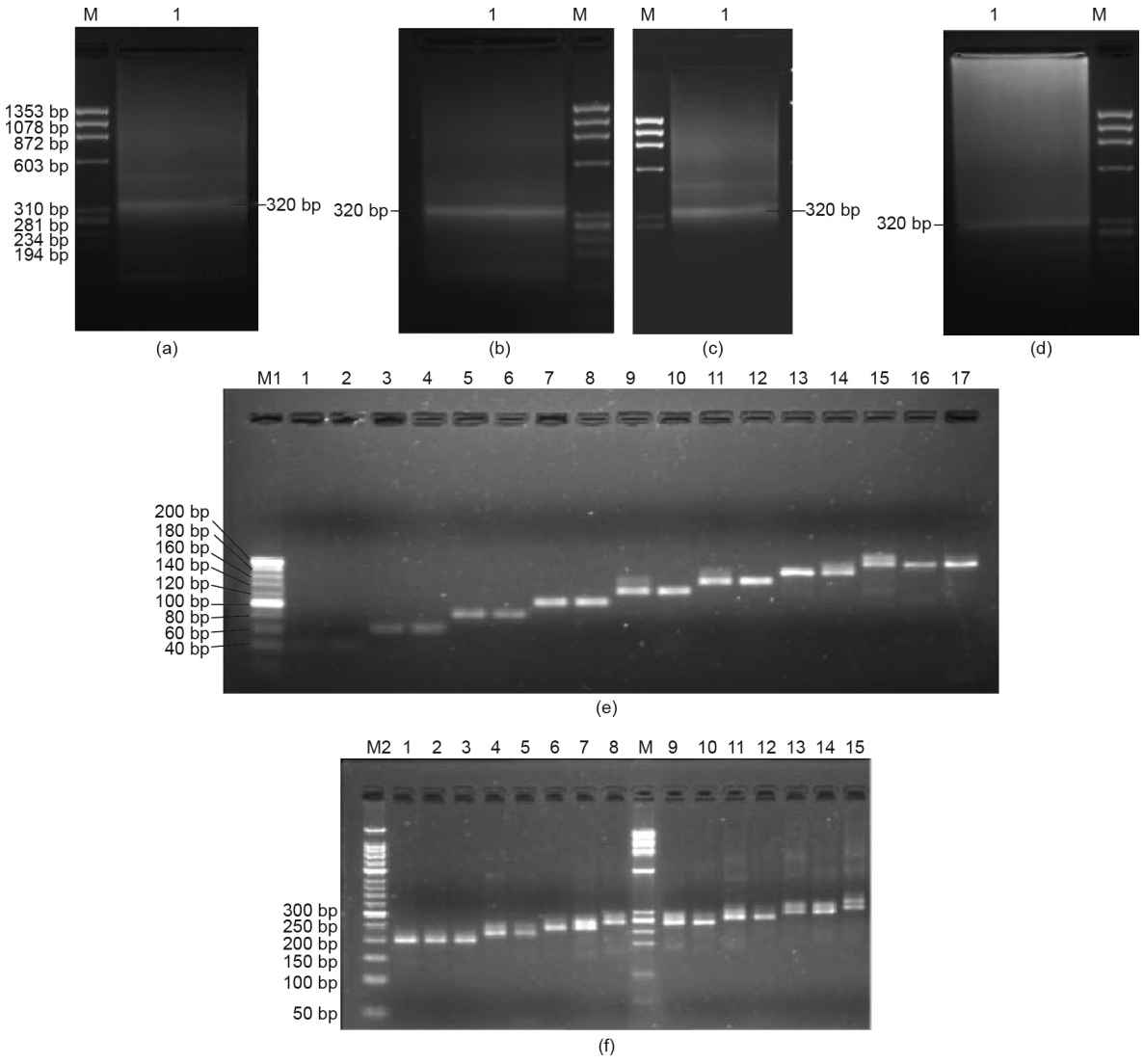

Fig. 8. Construction of the initial solution space. M, M1, and M2 are the molecular weight markers ΦX174-*Hae* III, 20 bp ladder, and 50 bp ladder, respectively. The 320 bp DNA bands of (a), (b), (c), and (d) are $S(G_1)$, $S(G_2)$, $S(G_3)$, and $S(G_4)$, respectively. (e) and (f) are the results of analysis of $S(G_1)$. Lanes 1–17 in (e) correspond to the primer pairs $\langle r_1, \overline{y_2}\rangle$, $\langle r_1, \overline{b_2}\rangle$, $\langle r_1, \overline{r_3}\rangle$, $\langle r_1, \overline{y_3}\rangle$, $\langle r_1, \overline{r_4}\rangle$, $\langle r_1, \overline{y_4}\rangle$, $\langle r_1, \overline{y_5}\rangle$, $\langle r_1, \overline{b_5}\rangle$, $\langle r_1, \overline{y_6}\rangle$, $\langle r_1, \overline{b_6}\rangle$, $\langle r_1, \overline{r_7}\rangle$, $\langle r_1, \overline{y_7}\rangle$, $\langle r_1, \overline{r_8}\rangle$, $\langle r_1, \overline{y_8}\rangle$, $\langle r_1, \overline{r_9}\rangle$, $\langle r_1, \overline{y_9}\rangle$, and $\langle r_1, \overline{b_9}\rangle$, respectively. Lanes 1–15 in (f) correspond to the primer pairs $\langle r_1, \overline{r_{10}}\rangle$, $\langle r_1, \overline{y_{10}}\rangle$, $\langle r_1, \overline{b_{10}}\rangle$, $\langle r_1, \overline{r_{11}}\rangle$, $\langle r_1, \overline{y_{11}}\rangle$, $\langle r_1, \overline{r_{12}}\rangle$, $\langle r_1, \overline{y_{12}}\rangle$, $\langle r_1, \overline{r_{13}}\rangle$, $\langle r_1, \overline{y_{13}}\rangle$, $\langle r_1, \overline{b_{13}}\rangle$, $\langle r_1, \overline{y_{14}}\rangle$, $\langle r_1, \overline{b_{14}}\rangle$, $\langle r_1, \overline{r_{15}}\rangle$, $\langle r_1, \overline{y_{15}}\rangle$, and $\langle r_1, \overline{b_{16}}\rangle$, respectively.

The first type was designed for the walk or circle, and parallel PCR was applied to find the true solutions. We give the steps using a path P = 16–20–22–26–30 of subgraph G_2 as an example.

First, the 50-fold dilutions of $S(G_2)$ were amplified by different primer pairs: $\langle b_{16}, \overline{r_{20}}\rangle$, $\langle b_{16}, \overline{y_{20}}\rangle$, $\langle r_{20}, \overline{y_{22}}\rangle$, $\langle y_{20}, \overline{r_{22}}\rangle$, $\langle r_{22}, \overline{y_{26}}\rangle$, $\langle y_{22}, \overline{r_{26}}\rangle$, $\langle r_{26}, \overline{y_{30}}\rangle$, $\langle r_{26}, \overline{b_{30}}\rangle$, $\langle y_{26}, \overline{b_{30}}\rangle$, $\langle y_{30}, \overline{r_{31}}\rangle$, and $\langle b_{30}, \overline{r_{31}}\rangle$. After PCR, five different DNA segments with different lengths were obtained. The DNA segments of 100 bp ($\langle b_{16}, \overline{x_{20}}\rangle$), 60 bp ($\langle x_{20}, \overline{x_{22}}\rangle$), 100 bp ($\langle x_{22}, \overline{x_{26}}\rangle$), 100 bp ($\langle x_{26}, \overline{x_{30}}\rangle$), and 40 bp ($\langle x_{30}, \overline{x_{31}}\rangle$) are shown in Fig. 9(a) and (b), where $x_i \in C_{v_i}$, i = 16, 17, …, 31. Lane 5 in Fig. 9(a) shows no product with the primer pair $\langle r_{22}, \overline{y_{26}}\rangle$, which means that there is no DNA strand with r_{22} and y_{26} in the initial solution space. This is the first PCR.

Next, the second PCR was run using different templates and different primer pairs, and the five DNA segments were combined to form three segments. Here T_i (i = 1, 2, 3) is used to represent the different templates. T_1 represents the mixtures of PCR products from Lanes 1 and 3 in Fig. 9, T_2 represents the mixtures from Lanes 1 and 4 in Fig. 9(b) as the template, and T_3 represents the mixtures from Lanes 2 and 5 in Fig. 9(b). With the primer pairs $\langle b_{16}, \overline{y_{22}}\rangle$, $\langle r_{26}, \overline{r_{31}}\rangle$, and $\langle r_{26}, \overline{r_{31}}\rangle$, three segments with 140 bp, 120 bp, and 120 bp, respectively, were acquired. The results are shown in Fig. 9(c).

Subsequently, the third PCR was run, and a 220 bp band was acquired with the primer pair $\langle b_{16}, \overline{r_{26}}\rangle$ by using the mixtures of PCR products from Lane 6 in Fig. 9(a) and Lane 1 in Fig. 9(c). The result is shown in Fig. 9(d).

Finally, the mixtures from Lane 1 in Fig. 9(d) and Lane 2 in Fig. 9(c) and the mixtures from Lane 1 in Fig. 9(d) and Lane 3 in Fig. 9(c) were amplified separately by primer pair $\langle b_{16}, \overline{r_{31}}\rangle$, and two different 320 bp bands were acquired (see Fig. 9(e)). Thus, at the end of the PCR operations, two sets consisting of 320 bp DNA sequences were obtained. In these sets, the coloring of vertices 20, 22, 26, and 30 were ascertained as follows: $b_{16}r_{20}y_{22}r_{26}y_{30}r_{31}$ and $b_{16}r_{20}y_{22}r_{26}b_{30}r_{31}$. These steps are shown in Fig. 9(f).

The second type was designed for the single edge whose two end vertices colors were undetermined. Here, we take $e = \{4, 8\}$ in subgraph G_1 as an example.

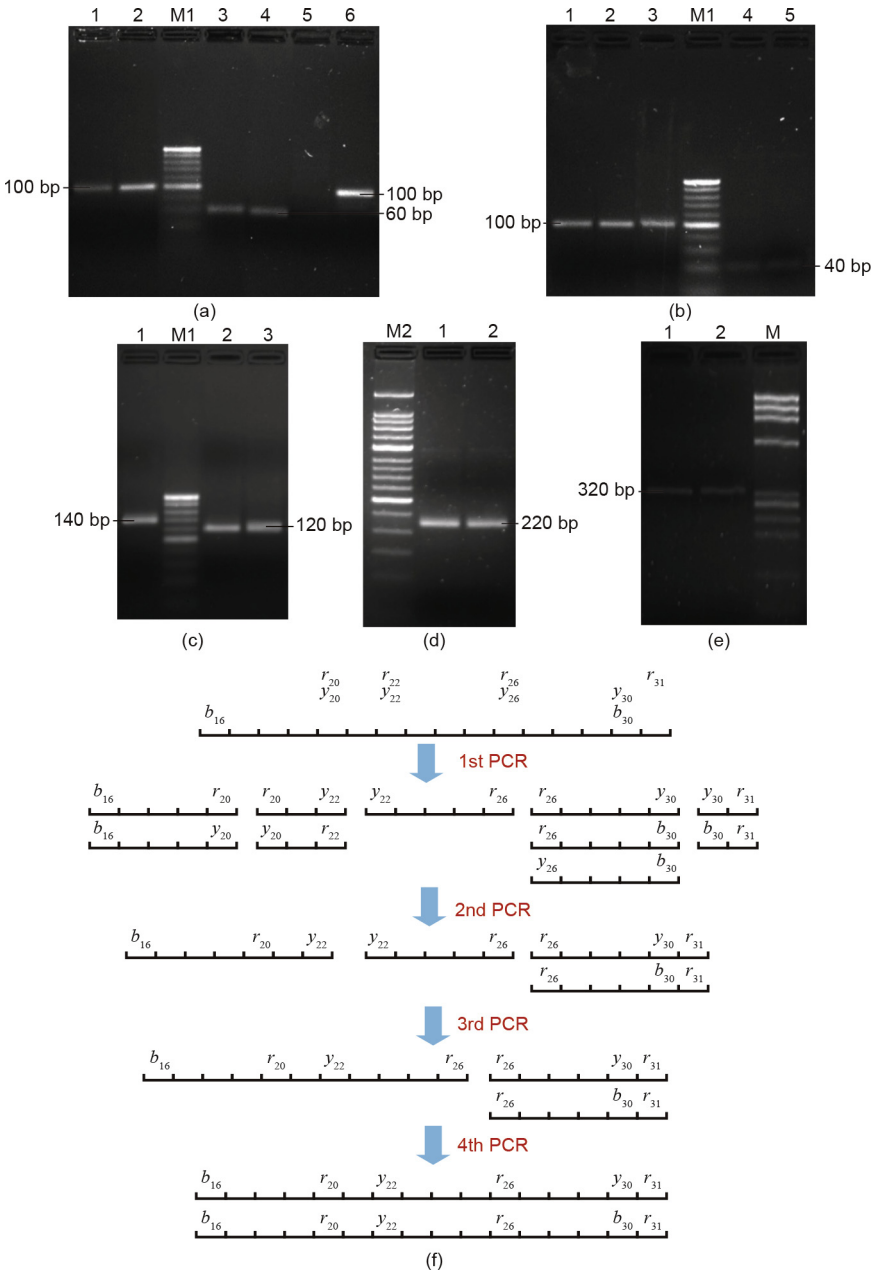

Fig. 9. PCR results. (a) Lane 1 to Lane 6 correspond to primer pairs $\langle b_{16}, \overline{r_{20}} \rangle$, $\langle b_{16}, \overline{y_{20}} \rangle$, $\langle r_{20}, \overline{y_{22}} \rangle$, $\langle y_{20}, \overline{r_{22}} \rangle$, $\langle r_{22}, \overline{y_{26}} \rangle$, and $\langle y_{22}, \overline{r_{26}} \rangle$, respectively; (b) Lane 1 to Lane 5 correspond to primer pairs $\langle r_{26}, \overline{y_{30}} \rangle$, $\langle r_{26}, \overline{b_{30}} \rangle$, $\langle y_{26}, \overline{b_{30}} \rangle$, $\langle y_{30}, \overline{r_{31}} \rangle$, and $\langle b_{30}, \overline{r_{31}} \rangle$, respectively; (c) Lane 1 to Lane 3 correspond to primer pairs $\langle b_{16}, \overline{y_{22}} \rangle$, $\langle r_{26}, \overline{r_{31}} \rangle$, and $\langle r_{26}, \overline{r_{31}} \rangle$, respectively; (d) Lane 1 and Lane 2 correspond to primer pair $\langle b_{16}, \overline{r_{26}} \rangle$; (e) Lanes 1 and 2 correspond to primer pair $\langle b_{16}, \overline{r_{31}} \rangle$; (f) an illustration of parallel PCR operations.

At first, the PCR products of the third cycle of subgraph G_1 were amplified by the primer pairs $\langle r_1, \overline{r_4} \rangle$, $\langle r_4, \overline{y_8} \rangle$, $\langle y_8, \overline{b_{16}} \rangle$, $\langle r_1, \overline{y_4} \rangle$, $\langle y_4, \overline{r_8} \rangle$, and $\langle r_8, \overline{b_{16}} \rangle$, and the 320 bp full-length DNA strands were divided into sequences of 80 bp ($\langle r_1, \overline{x_4} \rangle$), 100 bp ($\langle x_4, \overline{x_8} \rangle$), and 180 bp ($\langle x_8, \overline{b_{16}} \rangle$), respectively, where ($x_i = C_{v_i}, i = 1, 2, \ldots, 16$). The results are shown in Fig. 10(a); there is no product for $\langle y_4, \overline{r_8} \rangle$.

Next, PCR was run with primer pair $\langle r_1, \overline{y_8} \rangle$, using the mixtures of PCR products from Lanes 1 and 3 in Fig. 10(a) as a template, and a 160 bp band was acquired (see Fig. 10(b)).

At last, the mixtures of PCR products from Lane 5 in Fig. 10(a) and Lane 1 in Fig. 10(b) were amplified by primer pair $\langle r_1, \overline{b_{16}} \rangle$, and a 320 bp band was acquired. To test the color of v_4 and v_8, the 320 bp band was amplified by primer pairs $\langle r_1, \overline{r_8} \rangle$, $\langle r_4, \overline{r_8} \rangle$, $\langle y_4, \overline{r_8} \rangle$, and $\langle y_4, \overline{y_8} \rangle$, respectively. No product was obtained. Thus, the colors of vertices 4 and 8 were determined as r_4 and y_8.

The third type was designed to check the edges whose two end vertices colors had been determined. Here, $e = \{39, 42\}$ was used to illustrate the operation.

After finishing all PCRs of C = 34–38–42–46–34 and the first PCR of C = 34–38–42–46–34 of subgraph G_3, there were a total of six solutions (marked as (1) to (6) in Fig. 11), and the color of v_{39} and v_{42} could be determined as b_{39}, y_{39}, r_{42}, and y_{42}. In these cases, only the DNA strands with y_{39} and y_{42} are false solutions. To delete the false solutions, these six solutions were amplified by primer pair $\langle y_{39}, \overline{y_{42}} \rangle$ separately; only the false solutions would produce 80 bp bands after PCR. The results are shown in Fig. 11(a), and the principle is shown in Fig. 11(b).

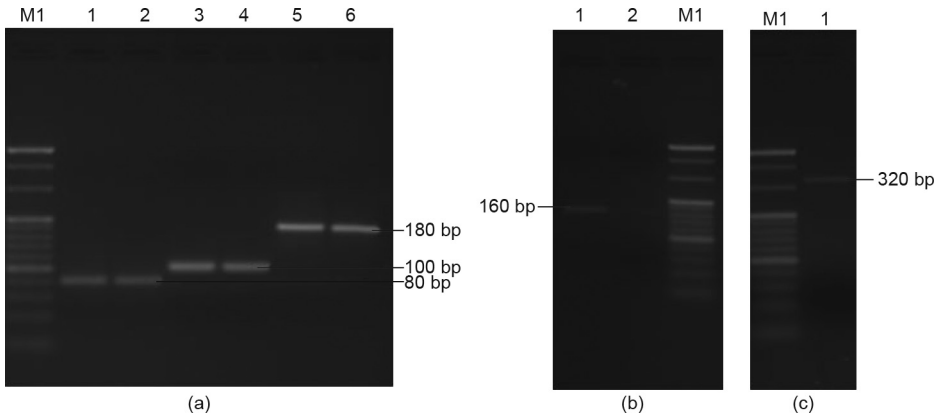

Fig. 10. PCR results for $e = \{4, 8\}$. (a) Lanes 1 and 2, 3 and 4, and 5 and 6 correspond to primer pairs $\langle r_1, \overline{r_4}\rangle$, $\langle r_4, \overline{y_8}\rangle$, and $\langle y_8, \overline{b_{16}}\rangle$; (b) Lanes 1 and 2 correspond to primer pair $\langle r_1, \overline{y_8}\rangle$; (c) Lane 1 corresponds to primer pair $\langle r_1, \overline{b_{16}}\rangle$.

4.5. Finding the true solution of graph G

To find the solutions of graph G, the solution spaces of the subgraphs must be combined, and then the false solutions must be deleted according to the operations described in Section 4.4.

First, the solution spaces mixtures of subgraphs G_1 and G_2 were amplified by primer pair $\langle r_1, \overline{r_{31}}\rangle$. The PCR results were DNA strands with a 620 bp band, which represented all possible three-colorings of $G[V_1 \cup V_2]$. Meanwhile, false solutions might also exist in this space, because there were five edges connecting G_1 and G_2. After deleting the false solutions, four true solutions of $G[V_1 \cup V_2]$ (X_1–X_4) were obtained (shown below). The results are shown in Fig. 12(a) and (b).

$$X_1 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31}$$

$$X_2 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31}$$

$$X_3 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 b_9 y_{10} r_{11} y_{12} r_{13} b_{14} y_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31}$$

$$X_4 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 b_9 y_{10} r_{11} y_{12} r_{13} b_{14} y_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31}$$

Similarly, the solution spaces mixtures of subgraphs G_3 and G_4 were amplified by primer pair $\langle r_{31}, \overline{y_{61}}\rangle$ to obtain the possible solutions of $G[V_3 \cup V_4]$. After deleting the false solutions, six true solutions Y_1–Y_6 remained (shown below). The results are shown in Fig. 12(c)–(e).

$$Y_1 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Y_2 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} r_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Y_3 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} r_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Y_4 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Y_5 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} r_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Y_6 = r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} r_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

Finally, the solution space of subgraphs $G[V_1 \cup V_2]$ and $G[V_3 \cup V_4]$ were combined by primer pair $\langle r_1, \overline{y_{61}}\rangle$. This primer pair only resulted in a band corresponding to 1220 bp. After deleting false solutions, eight true solutions Z_1–Z_8 were obtained (shown below). The results are shown in Fig. 13.

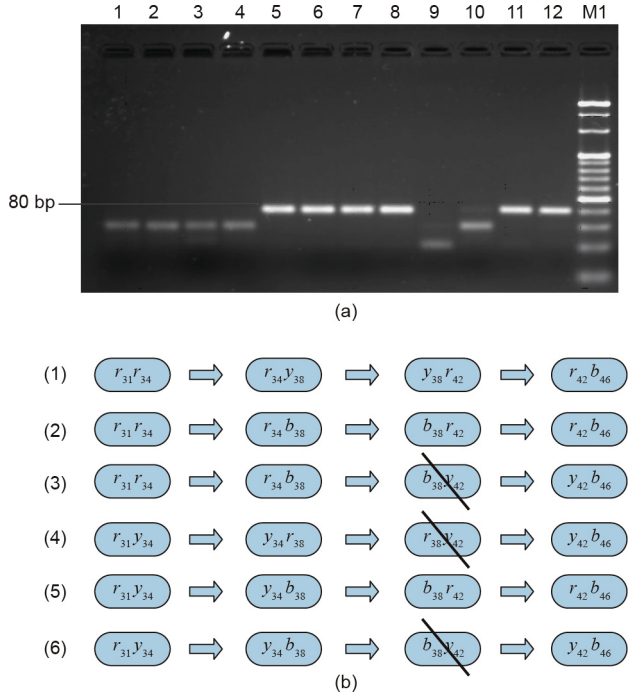

Fig. 11. PCR results for $e = \{39, 42\}$. (a) All lanes correspond to primer pair $\langle y_{39}, \overline{y_{42}}\rangle$: Lanes 1 and 2 correspond to solution (1), Lanes 3 and 4 correspond to solution (2), Lanes 5 and 6 correspond to solution (3), Lanes 7 and 8 correspond to solution (4), Lanes 9 and 10 correspond to solution (5), and Lanes 11 and 12 correspond to solution (6); (b) an illustration of deleting false solutions; "\" stands for deleting false solutions.

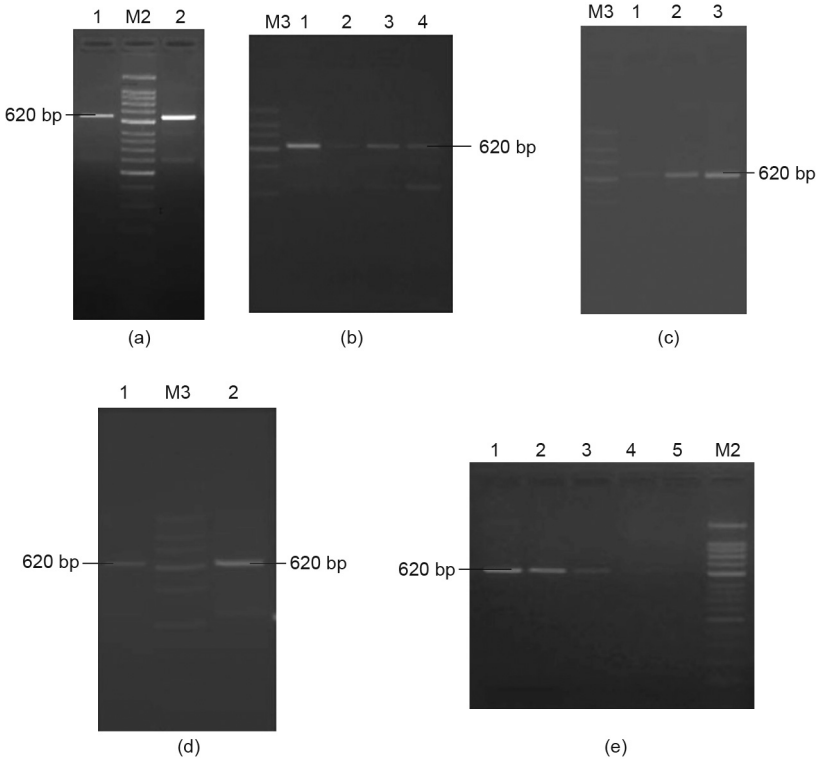

Fig. 12. The solutions for $G[V_1 \cup V_2]$. (a) Lanes 1 and 2 correspond to solutions X_1 and X_2, respectively; (b) Lanes 1 and 2 correspond to solution X_3, and Lanes 3 and 4 correspond to solution X_4; (c) Lanes 1 and 2 correspond to solution Y_1, and Lane 3 corresponds to solution Y_2; (d) Lane 1 corresponds to solution Y_3, and Lane 2 corresponds to solution Y_4; (e) Lane 1 corresponds to solution Y_5, Lane 2 corresponds to solution Y_6, Lane 3 corresponds to a positive control, and Lanes 4 and 5 correspond non-solution. All the solutions denoted as Y_j, where $j = 1, 2, 3, 4, 5, 6$, correspond to primer pair $\langle r_{31}, \overline{y_{61}} \rangle$.

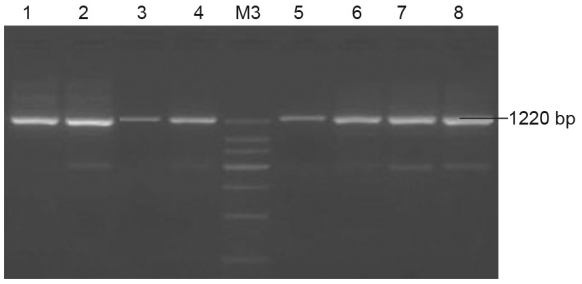

Fig. 13. The solutions for $G[V_3 \cup V_4]$. M3 is the DNA molecular weight marker 150 bp ladder. Lane 1 to Lane 8 correspond to solutions Z_1 to Z_8, respectively.

$$Z_1 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Z_2 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Z_3 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Z_4 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Z_5 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Z_6 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

$$Z_7 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} r_{60} y_{61}$$

$$Z_8 = r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} r_{57} b_{58} y_{59} b_{60} y_{61}$$

All eight solutions Z_1–Z_8 were purified and sequenced by Beijing AuGCT Biology Company. The sequences were analyzed by Gene Tools and improved to be the true solutions of graph G.

5. Complexity analysis

In this section, we analyze the complexity of this DNA computing model in two aspects: the complexity of the initial solution space and the complexity of the experiments. We provide a calculation formula to estimate the amount of initial solution space for typical and for worst cases.

5.1. Calculation of the walks in a given graph

To illustrate the complexity of the initial solution space, we first introduce a formula, which can be used to calculate the number of the walks between any two vertices.

Lemma 5.1 [28]**:** Let G be a graph with n vertices and $V(G) = \{v_1, v_2, \ldots, v_n\}$. A is the adjacent matrix of G. The number of walks of length l in G, from v_i to v_j, is the entry in position (i,j) of the matrix A^l.

5.2. Complexity analysis of decreasing the initial solution space

In previous DNA computing models for solving NP-complete problems, a large number of DNA molecules were frequently used to encode and enumerate an infinite amount of data. However, those models rarely considered decreasing false solutions by constructing the initial solution space; rather, they used the enumerating method. In fact, this enumerating idea does not work for large-scale problems for two reasons. First, with an increase of problem scale, the number of DNA molecules becomes too large to obtain. Second, even if the number of DNA molecules can be designed, the experiments will be too difficult to implement due to the larger size and greater number of molecules. Thus, it is necessary to find more efficient methods to decrease the initial solution space when designing a DNA computing model to solve NP problems. In this article, we designed a DNA computing model for the coloring problem by using an optimization idea to construct an initial solution space, rather than by enumerating.

Theorem 5.2: Suppose that G is a graph with $|V(G)| = n$ and G_1 is its subgraph, as proposed by Section 3.2, where $V(G) = \{v_1, v_2, \ldots, v_n\}$, with the bridge vertices v_1 and v_t. Let $B(G_1)$ be the probe graph of G_1. Then the number of DNA sequences representing all possible coloring solutions in the initial solution space to G_1 is equal to the number of paths connecting the bridge vertices v_1 and v_t with length $t-1$ in $B(G_1)$, denoted by $N_p(v_1, v_t)$, which is given by the following formula:

$$N_p(v_1, v_t) = A^{t-1}[B(G_1)](1, t-1) \tag{5}$$

Proof: The vertex x_i in 99.9997% represents a coloring of the vertex v_i in G_1 by the definition of $B(G_1)$. The edge $\{x_i, z_{i+1}\} \in B(G_1)$ represents the probe between x_i and z_{i+1}. That is, when the color of v_i is x, the color of v_{i+1} can be chosen as z, where $x, z \in \{r, y, b\}$. Consequently, each path from vertex r_1 to vertex b_t denotes one possible coloring of subgraph G_1. Furthermore, all the paths between r_1 and b_t in $B(G_1)$ represent all the possible solutions of the initial solution space.

Considering the structure of the probe graph $B(G_1)$, we divide $V(B(G_1))$ into v_t parts, where the vertex with subscript i is the ith part. It is concluded that there must at least exist an edge connecting the ith part and the $(i+1)$th part by the definition of a probe graph. Thus, the length of the paths between r_1 and b_t in $B(G_1)$ is only $t-1$. And each walk from r_1 to b_t is just the path between r_1 and b_t. Thus, the theorem is proved by Lemma 5.1. □

Corollary 5.3: The number of DNA strands in initial solution spaces $S(G_1)$, $S(G_2)$, $S(G_3)$, and $S(G_4)$ of the subgraphs G_1, G_2, G_3, and G_4 shown in Fig. 4 are 89, 81, 412, and 151, respectively. The corresponding percentage of DNA strands deleted in the enumerable initial solution spaces are 99.9998%, 99.9998%, 99.999%, and 99.9997%, respectively.

Proof: Notice that vertices v_1 and v_{16} are the bridge vertices of subgraph G_1. Thus, the number of DNA strands in $S(G_1)$ is $N_{33}(r_1, b_{16})$, which is the number of paths with length 15 from vertex r_1 to b_{16} in B_1 (see Fig. 7) by Theorem 5.2.

The adjacent matrix $A(B_1)$ of B_1 is

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

The correspondence between the vertices of $A(B_1)$ and B_1 is

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ r_1 & y_2 & b_2 & r_3 & y_3 & r_4 & y_4 & y_5 & b_5 & y_6 & b_6 & r_7 & y_7 & r_8 & y_8 & r_9 & y_9 \\ 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & \\ b_9 & r_{10} & y_{10} & b_{10} & r_{11} & y_{11} & r_{12} & y_{12} & r_{13} & y_{13} & b_{13} & y_{14} & b_{14} & r_{15} & y_{15} & b_{16} & \end{pmatrix}$$

According to Lemma 5.1, we have $A(B_1)^{15}(1,33)=89$, and $\frac{89}{43046721}\approx 0.000002=0.0002\%$. The result shows that 99.9998% of the false solutions are deleted when constructing the initial solution space $S(G_1)$.

Similarly, we have $A(B_2)^{15}(1,32)=81$, and $\frac{81}{43046721}\approx 0.00000188=0.000188\%$ for subgraph G_2; $A(B_3)^{15}(1,35)=412$, and $\frac{412}{43046721}\approx 0.0000096=0.00096\%$ for subgraph G_3; and $A(B_4)^{15}(1,32)=151$, and $\frac{151}{43046721}\approx 0.0000035=0.00035\%$ for subgraph G_4, respectively. The results show that 99.9998%, 99.999%, and 99.9997% of the false solutions were deleted when constructing the initial solution spaces $S(G_2)$, $S(G_3)$, and $S(G_4)$, respectively. □

5.3. Reducing operation complexity by parallel PCR technology

In our experiment, false solutions are deleted by designed parallel PCR operation on the forward paths (see Section 3.7). In this way, the bio-operation times are greatly reduced. The bio-operation complexity is analyzed in this section.

Let $P=v_{i0}v_{i1}\ldots v_{im}$ with $|V(P)|=m+1$ be a forward path in subgraph G_1 with bridge vertices v_1 and v_t. Then we must consider the relationships between the two bridge vertices and path $P=v_{i0}v_{i1}\ldots v_{im}$ with three cases when we design corresponding parallel PCR operation:

Case 1: There are no bridge vertices in $V(P)$. That is, $v_{i0}\neq v_1$ and $v_{im}\neq v_t$. For this case, we need to add the two bridge vertices so as to analyze the advantages of the parallel PCR operations (see Fig. 14(a)).

Case 2: There is one bridge vertex in $V(P)$. That is, $v_{i0}=v_1$ but $v_{im}\neq v_t$, or $v_{i0}\neq v_1$ but $v_{im}=v_t$. For this case, we need to add the one bridge vertex v_1 or v_t (see Fig. 14(b)).

Case 3: There are two bridge vertices in $V(P)$. That is, $v_{i0}=v_1$ and $v_{im}=v_t$. For this case, we need only to consider the ordering vertices in path P (see Fig. 14(c)).

Since the color of the bridge vertex v_1 (or v_t) is fixed in advance, its adjacent vertices in Cases 2 and 3 are taken as the other colors aside from the color of v_1 (or v_t). Since there are three cases shown in Fig. 14, we will only discuses Case 1, because the other two cases can be treated as special cases of Case 1. Let P belongs to Case 1, then $P'=P-\{v_1,v_{i0}\}$ or $P'=P-\{v_{im},v_t\}$ should belongs to Case 2, and $P''=P-\{v_1,v_{i0}\}-\{v_{im},v_t\}$ will be Case 3.

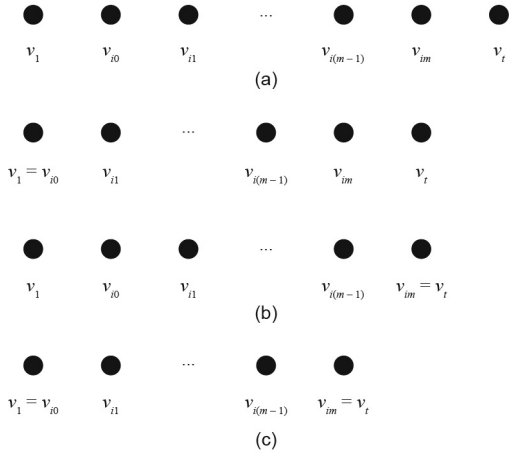

Fig. 14. Three cases between the two bridge vertices and a forward path. (a) $v_{i0}\neq v_1$ and $v_{im}\neq v_t$; (b) $v_{i0}=v_1$ but $v_{im}\neq v_t$, or $v_{i0}\neq v_1$ but $v_{im}=v_t$; (c) $v_{i0}=v_1$ and $v_{im}=v_t$.

Next, let us discuss the parallel PCR method. First, we determine the number and sequence of the vertices for the forward path $P=v_{i0}v_{i1}\ldots v_{im}$ using the approach given in Fig. 14. Next, we perform the PCR operations. In practice, we do several times of PCR to fulfill one deleting false solutions operation. We found that the times of PCR operation are only relevant to the number of vertices or edges. Taking the first case above as an example, it is necessary to amplify the strands with the primer pairs $\langle v_1,v_{i0}\rangle$, $\langle v_{i0},v_{il}\rangle,\ldots\langle v_{i(m-1)},v_{im}\rangle$, and $\langle v_{im},v_t\rangle$ in the first PCR operation. The second PCR operation is designed to connect the consecutive two segments of the results of the previous PCR operation. The following PCR processes perform similar operations to the second PCR operation. In Fig. 15, we refer to cases with five vertices (a path with two edges) and six vertices (a path with three edges) as examples in order to illustrate the process of parallel PCR, where the edges mean how we do the PCR operation to connect the DNA segments to get a completed DNA sequence.

It is concluded from Fig. 15 that the number of PCR operation is 3 for the case of the path with two edges, and for the case of the path with six edges the number of PCR operation is 4. Thus, we have Theorem 5.4.

Theorem 5.4: Suppose that P is the forward path of a subgraph with x edges. If x satisfies $2^{l-1}<x\leqslant 2^l$, we define $\langle x\rangle=2^l$. Let the number of PCR operation be represented as PCR(x). Then, it can be concluded that

$$\mathrm{PCR}(x)=1+\log_2\langle x+2\rangle,\quad \text{where } x\geqslant 1 \tag{6}$$

Proof: First, we consider the case $x+2=2^l$. Suppose that the number of edges in path P is x. That is, the number of vertices in path P is $x+1$. We already analyze the three cases between the two bridge vertices in a forward path in Fig. 14, and Cases 2 and 3 are the special cased of Case 1. So without loss of generality, we focus our attention on Case 1.

Since there are no bridge vertices in path P, it is necessary to increase two bridge vertices according to the requirements of the PCR operations, such that the number of vertices increases to $x+3$. Then, we obtain the following equation:

$$x+3=2^l+1$$

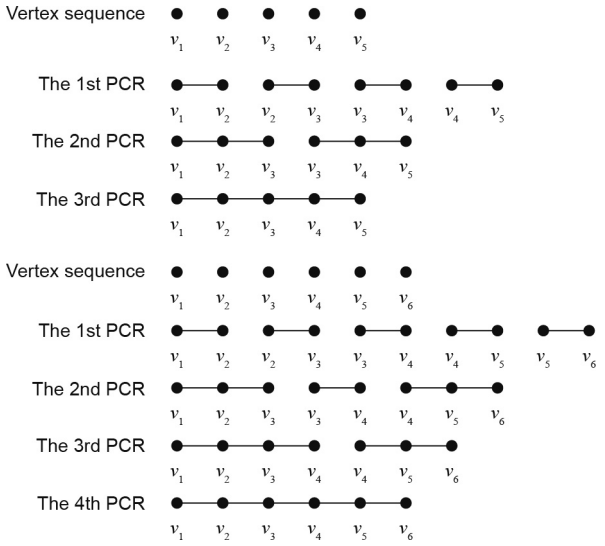

Fig. 15. An illustration of parallel PCR technology.

As a result, the number of primer pairs of the first PCR can be concluded as shown below:

$$(x+3)-1=x+2=2^l$$

Following the above operations, the primer pair number in the second PCR is 2^{l-1}. In general, the primer number in the lth PCR is concluded to be:

$$\mathrm{PCR}(x)=2^{l-(l-1)}=2$$

Thus, the PCR can be accomplished in the $(l+1)$th time. In other words, the number of PCR operation required are $l+1$ when $x+2=2^l$. Then, we obtain the following mathematical relation:

$$\mathrm{PCR}(x)=l+1=1+\log_2 2^l=1+\log_2\langle 2^l\rangle=1+\log_2\langle x+2\rangle$$

Similarly, if $x+2=2^{l+1}$, then the number of PCR operation satisfy the formula as follows:

$$\mathrm{PCR}(x)=l+2=1+\log_2\langle 2^{l+1}\rangle=1+\log_2\langle x+2\rangle$$

Here, we demonstrate the case $x+2=2^l+1$:

As shown above, the number of vertices in path P is $x+3$ by increasing two bridge vertices. Thus, when $x+2=2^l+1$, there are 2^l+2 vertices. The number of primer pairs in the first PCR is 2^l+1, as induced by the following formula: $(x+3)-1=x+2=2^l+1$.

The number of primer pairs in the second PCR is then $2^{l-1}+1$. Similarly, the number of primer pairs in the $(l+1)$th PCR is 2. The $(l+2)$th PCR can be completed.

On the other hand, we can obtain the number of the PCR operation as follows:

$$\begin{aligned}\mathrm{PCR}(x)&=l+2=1+(l+1)=1+\log_2 2^{l+1}=1+\log_2\langle 2^l+1\rangle\\&=1+\log_2\langle x+2\rangle\end{aligned}$$

Thus, the conclusion above is verified when $x+2=2^l+1$.

For $2\leqslant y<2^l$, the number of PCR operation when $x+2=2^l+y$ is more than that of the case $x+2=2^{l+1}$. The number of PCR operation for these two cases above is $l+2$. Thus, the conclusion holds when $2\leqslant y<2^l$ and $x+2=2^l+y$. Consequently, Theorem 5.4 is demonstrated. □

Theorem 5.4 fully shows the relations between the length x of forward path P and the number of the PCR operation. Of course, the number can be reduced by parallel PCR operations, which will greatly accelerate the computation speed. We can now see the advantages of parallel PCR operations, as shown in Table 2 and Fig. 16.

Corollary 5.5: Let P be the forward path in a graph G_1 with x edges. It can be induced by using Theorem 5.4 that

$$\lim_{x\to\infty}\frac{\log_2\langle x+2\rangle+1}{3x}=0$$

Proof: Let $x+2=2^l+y$, where $1\leq y\leq 2^l$. Then, we have

$$\langle x+2\rangle=2^{l+1}=2(2^l+y-y)<2(2^l+y)$$

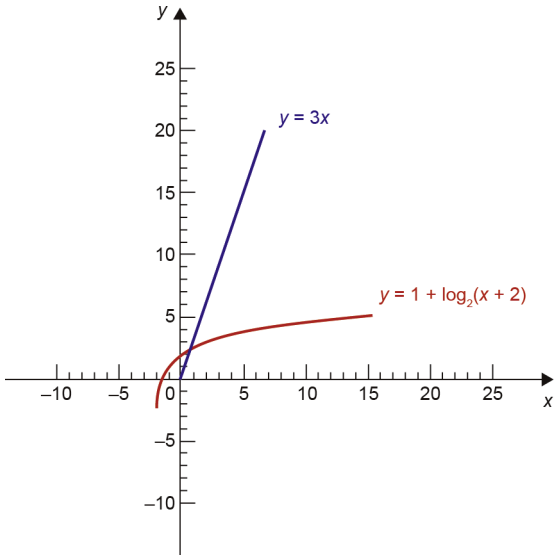

Fig. 16. A comparison between parallel PCR and general PCR operations for a single edge.

In addition, since y is less than 2^l, then it is induced that $2^{l+1}>y+2^l$. Consequently, the following consecutive inequalities hold:

$$y+2^l<2^{l+1}<2(2^l+y)$$

That is, x satisfies the relation below:

$$x+2<\langle x+2\rangle<2(x+2)$$

Then, we obtain

$$\log_2(x+2)<\log_2\langle x+2\rangle<\log_2 2(x+2),\quad\text{and}$$

$$\frac{1+\log_2(x+2)}{3x}<\frac{1+\log_2\langle x+2\rangle}{3x}<\frac{1+\log_2 2(x+2)}{3x}$$

On the basis of L'Hospital's rule, we obtain

$$\lim_{a\to\infty}\frac{\log_2(x+2)+1}{3x}=\lim_{a\to\infty}\frac{\log_2 2(x+2)+1}{3x}=0$$

Therefore, the formula $\lim_{x\to\infty}\frac{\log_2\langle x+2\rangle+1}{3x}=0$ is true. □

6. Conclusions

Since shortly after the discovery of the double helix structure of DNA, the idea of using DNA molecules to process information has been a dream, one that originated from Feynman's concept of constructing submicroscopic computers [1]. This dream was

Table 2
A comparison between parallel PCR operations and general PCR operations for a single edge.

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PCR(x)	3	3	4	4	4	4	5	5	5	5	5	5	5	5	6
$3x$	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
PCR(x)	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
$3x$	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90

x is the number of edges, and $3x$ and PCR(x) denote the number of operation of general PCR and parallel PCR for a single edge, respectively.

ultimately fulfilled by Adleman in 1994, when he established a DNA computing model for a Hamiltonian graph with seven vertices. These works greatly contributed to the development of DNA computing.

The success of our DNA computing model not only demonstrates its huge computational capability, but also provides a seemingly feasible method to solve the exponential explosion problem, which may be the biggest bottleneck preventing DNA computing from solving larger-scale NP problems. In this paper, a graph vertex coloring problem with a computing complexity of $O(3^{59})$ is solved by using the advantages of the DNA computing model: the construction of an un-enumerating initial solution space, the division and combination of subgraphs, and parallel bio-operation.

By dividing subgraphs, reducing the vertex colors without losing solutions, and ordering the vertices in subgraphs, the exponential explosion problem is effectively avoided. According to Lemma 5.1, the initial solution space of each subgraph is 89, 81, 412, and 151, respectively, all of which are much lower than 3^{16} (the worst complexity). Therefore, the values of DNA sequences in the un-enumerating initial solution space are 99.9998%, 99.9998%, 99.999%, and 99.9997% lower than those in an enumerating initial solution space.

In this model, we also improve the computing efficiency by dividing the graph into four subgraphs and simultaneously deleting the false solutions of these subgraphs. After obtaining the solutions of each subgraph, the true solutions are found by combining the subgraphs. It is possible that the ability of this model to solve the graph vertex coloring problem will reach 100 vertices.

In this model, false solutions are deleted according to the path rather than the edge. Thus, all edges in one path are dealt with at the same time. In this way, the number of PCR operation required decreases. The efficiency of this DNA computing model is greatly improved by a significant innovation: the "parallel deleting method."

In general, many researchers use PCR for amplification with only two primers and one template, because they think that PCR is not very reliable with many templates. However, in this model, we use PCR to amplify all templates in a test-tube. In order to obtain reliable results, the primer pairs are strictly designed according to the constraints of PCR, and the reaction conditions and the concentration of the components are optimized for each PCR. Only the bands representing the specific amplification are cut and purified for the next operation. In addition, contrast tests are implemented to avoid false positives and false negatives in the experiments. The sequencing shows that the results of the experiments are reliable. Because of its reliability and its ability to reduce the solution space, we assert that this DNA computing model can be used to solve larger-size problems, and that DNA molecular tools can be used to carry out larger-scale NP problems. In future, it may be possible to use DNA computing for certain practical engineering problems.

Acknowledgements

The authors are grateful for the support from the National Natural Science Foundation of China (61632002, 61379059, and 61572046).

Compliance with ethics guidelines

Jin Xu, Xiaoli Qiang, Kai Zhang, Cheng Zhang, and Jing Yang declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Feynman RP. Minaturization. New York: Reinhold; 1961.
[2] Adleman LM. Molecular computation of solutions to combinatorial problems. Science 1994;266(5187):1021–4.
[3] Lipton RJ. DNA solution of hard computational problems. Science 1995;268 (5210):542–5.
[4] Ouyang Q, Kaplan PD, Liu S, Libchaber A. DNA solution of the maximal clique problem. Science 1997;278(5337):446–9.
[5] Sakamoto K, Gouzu H, Komiya K, Kiga D, Yokoyama S, Yokomori T, et al. Molecular computation by DNA hairpin formation. Science 2000;288 (5469):1223–6.
[6] Rothemund PWK. A DNA and restriction enzyme implementation of turing machines. In: Lipton RJ, Baum EB, editors. DNA based computers. Providence: American Mathematical Society; 1995. p. 75–119.
[7] Benenson Y, Paz-Elizur T, Adar R, Keinan E, Livneh Z, Shapiro E. Programmable and autonomous computing machine made of biomolecules. Nature 2001;414 (6862):430–4.
[8] Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. Nature 2004;429 (6990):423–9.
[9] Winfree E. Algorithmic self-assembly of DNA [dissertation]. Pasadena: California Institute of Technology; 1998.
[10] Braich RS, Chelyapov N, Johnson C, Rothemund PW, Adleman L. Solution of a 20-variable 3-SAT problem on a DNA computer. Science 2002;296 (5567):499–502.
[11] Berge C, Minieka E. Graphs and hypergraphs. New York: Elsevier Science Publishing Co, Inc.; 1973.
[12] Briggs P, Cooper KD, Dennedy K, Torczon L. Coloring heuristics for register allocation. In: Proceedings of the ACM SIGPLAN 1989 conference on programming language design and implementation. 1989 Jun 19–23; Portland, OR, USA; 1989. p. 275–84.
[13] Chaitin GJ. Register allocation & spilling via graph coloring. In: Proceedings of the ACM SIGPLAN 1982 conference on compiler construction; 1982 Jun 23–25; Boston, MA, USA; 1982. p. 98–105.
[14] Johnson DS. Worst case behavior of graph coloring algorithm. In: Proceedings of the 5th southeastern conference on combinatorics, graph theory and computing; 1974 Feb 25–Mar 1; Boca Raton, FL, USA; 1974. p. 513–27.
[15] Blum A. New approximation algorithms for graph coloring. J Assoc Comput Mach 1994;41(3):470–516.
[16] Karger D, Motwani R, Sudan M. Approximate graph coloring by semi-definite programming. J Assoc Comput Mach 1998;45(2):246–65.
[17] Schiermeyer I. Deciding 3-colorability in less than $O(1.415n)$ steps. In: Proceedings of the 19th International Workshop on Graph-Theoretic Concepts in Computer Science; 1993 Jun 16–18; Utrecht, The Netherlands. London: Springer; 1993. p. 177–88.
[18] Beigel R, Eppstein D. 3-coloring in time $O(1.3289n)$. J Algorithms 2005;54 (2):168–204.
[19] Jonoska N, Karl SA, Saito M. Three dimensional DNA structures in computing. Biosystems 1999;52(1–3):143–53.
[20] Jonoska N, Sa-Ardyen P, Seeman NC. Computation by self-assembly of DNA graphs. Genet Program Evolvable Mach 2003;4(2):123–37.
[21] Sa-Ardyen P, Jonoska N, Seeman NC. Self-assembling DNA graphs. Nat Comput 2003;2(4):427–38.
[22] Liu W, Zhang F, Xu J. A DNA algorithm for the graph coloring problem. J Chem Inf Comput Sci 2002;42(5):1176–8.
[23] Gao L, Xu J. A DNA algorithm for graph vertex coloring problem. Acta Electronic Sinica 2003;31:494–6.
[24] Xu J, Qiang X, Fang G, Zhou K. A DNA computer model for solving vertex coloring problem. Chin Sci Bull 2006;51(20):2541–9.
[25] Garey MR, Johnson DS. Computers and intractability: A guide to the theory of NP-completeness. New York: W. H. Freeman & Co.; 1979.
[26] Bondy JA, Murty USR. Graph theory with applications. New York: American Elsevier Publishing Company, Inc.; 1976.
[27] Zhang K, Pan L, Xu J. A global heuristically search algorithm for DNA encoding. Prog Nat Sci 2007;17(6):745–9.
[28] Biggs N. Algebraic graph theory. 2nd ed. Cambridge: Cambridge University Press; 1993.