



ELSEVIER

Contents lists available at ScienceDirect

Engineering

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



Research
Cybersecurity—Article

基于探针图的并行型图顶点着色 DNA 计算模型

许进^{a,*}, 强小利^b, 张凯^c, 张成^a, 杨静^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

关键词

DNA 计算

图顶点着色问题

聚合酶链反应 (PCR)

摘要

目前 DNA 计算机研究中遇到的最大瓶颈是解空间指数爆炸问题,即随着问题规模的增大,所需要作为信息处理“数据”的 DNA 分子呈指数级增大。本文提出了一种新颖的图顶点着色 DNA 计算模型,该模型正是围绕着如何克服解空间指数爆炸问题以及如何提高运行速度而设计的。其主要贡献有:①通过如下三种方法来克服解空间指数爆炸问题:顶点颜色集的确定方法;子图分解方法以及子图中的顶点优化排序方法;②设计了一种并行型聚合酶链反应 (PCR) 操作技术,应用这种技术一次可以对图中多条边来删除非解,使得生物操作次数大大减少,极大地提高了运行速度。本文以一个 3-着色的 61 个顶点的图为例,实验表明,99% 的非可行解在构建初始解空间时就被删除,并利用 DNA 自组装和并行 PCR 方法,通过识别、拼接以及组装等技术得到解。由于该模型对任意 61 个顶点的图是同样的操作方法,这就意味着,该模型的搜索能力可以达到 $O(3^{39})$ 。

© 2018 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. 引言

1961年, Feynman提出了分子计算的构想[1], 33年后, Adleman实现了Feynman的构想, 在1994年提出了以DNA分子作为“数据”, 以生物酶或生物操作为“工具”来进行信息处理的DNA计算模型[2]。从1994年至今, DNA计算模型无论在理论、实验还是应用等方面, 都取得了很大的进展。十多年的研究表明, 目前阻碍DNA计算机发展的最大瓶颈是: 在求解大规模的优化计算问题上, 特别是一些困难的NP (非确定性多项式)-完全问题上, 随着问题规模的增大, 所需要的DNA量呈指数增长。

1.1. 相关研究综述

1994年, Adleman[2]创立了以DNA分子为“数据”, 以生物酶和生物操作为信息处理“工具”的DNA计算模型。随后, 许多学者在此领域内作出了杰出的工作, 如Lipton[3]建立了SAT问题的DNA计算模型, Ouyang等[4]建立了求解图的最大团问题的DNA计算模型, Sakamoto等[5]建立的求解可满足性 (SAT) 问题的发卡DNA计算模型, Rothmund等[6]建立了图灵DNA计算机模型。除此之外, DNA分子建立用于基因表达逻辑控制的“自律式分子计算机”[7,8], 1998年Winfree等[9]提出了自组装DNA计算模型, 2002年Braich等[10]建立

* Corresponding author.

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

了具有20个变量3-SAT问题的DNA计算机模型，其搜索次数超过104万次。这是迄今为止用非电子工具所解决的规模最大的一个问题。

图的顶点着色问题是指对图中的每个顶点分配一种颜色，使得相邻的顶点着不同的颜色。这是一个典型的NP-完全问题，由于该问题在许多应用领域如排课表问题[11]、寄存器分配问题[12,13]等中都广泛存在，随着图的规模的扩大，计算量将会呈指数性增加，如何找到一个高效的算法就成为了解决有关问题的关键和突破点。1974年，Johnson[14]展开了对图染色问题的近似算法的分析工作。之后，Blum[15]提出用 $O(n^{3/8} \log^{8/5} n)$ 种颜色可以对具有 n 个顶点的3-可着色图进行染色。Karger等[16]提出了一种随机多项式时间算法，利用该算法可以对3-可着色图用 $\min\{O(\Delta^{1/3} \log^{1/2} \Delta \log n), O(n^{1/4} \log^{1/2} n)\}$ 种颜色进行顶点染色。Schiermeyer[17]给出了一种复杂算法，利用该算法，可以在小于 $O(1.415^n)$ 的时间内，给出3-着色方案。之后，Beigel和Eppstein[18]改进了算法，建立了一种快速算法，使得可以在 $O(1.3278^n)$ 时间内求解图的3-着色问题。

基于DNA计算模型的图顶点着色方面的研究主要有如下工作：1999年，Jonoska等[19]提出将3D的DNA分子用于DNA计算，在理论上证明，3D的DNA分子可以形成一个图当且仅当这个问题的解存在。在上述工作基础上，Jonoska和Sa-ardyen等利用生化操作构造了3-臂和4-臂的DNA分子，并将其用于求解顶点3-着色问题[20,21]。Liu等[22]和Gao等[23]先后分别提出了利用酶切操作技术来删除非解的用于求解图顶点着色的DNA计算模型。Xu等[24]建立了一种利用磁珠分离技术来删除非解的图顶点着色DNA计算模型，并通过实验验证了该模型。

1.2. 本文的工作

围绕着如何克服解空间指数爆炸问题与如何提高计算过程中的运算速度，本文建立了一种新颖的图顶点着色DNA计算模型，称为并行型图顶点着色DNA计算模型。该模型适用于求解任意着色的图顶点着色问题。文中以展开分析研究，这些方法与技术自然可以推广到任意情形。

对于规模较大的图而言，本模型的基本思想是：首先将一个给定的图分解成若干个子图，目的是不仅使得生物操作易于实现，而且将尽可能多的非解较早地给予删除，这样可为后面的生化操作方便；然后对较小的每

个子图求解，该求解过程涉及诸如顶点排序、桥点（即在子图中一对度数为最大的相邻顶点）的确定、每个顶点颜色集的确定、DNA序列编码、探针的确定、初始解空间的建立、非解的删除等（具体见3.3~3.7节）；最后，合并各个子图并逐步删除非解，直到合成为原始图（详见3.8节）。

本文的主要贡献有：①通过如下三种方法来克服解空间指数爆炸问题：子图分解方法、减少顶点颜色集的方法以及通过子图顶点排序方法来促使构造尽可能多的探针方法；②设计了一种并行型聚合酶链反应（PCR）操作技术，应用这种技术一次可以对图中多条边进行非解删除，使得生物操作次数大大减少，极大地提高了运行速度，并给出了提高运行速度相应的数学模型。

利用该计算模型，我们对如图1所示的61个顶点的图进行了实验计算，得到了满足该图正常3-着色的所有8个解。可以证明当 v_1 和 v_n 给定之后，本模型的计算能力可以达到 $O(3^{59})$ 。

1.3. 本文的大纲

本文安排如下：第二部分给出文章所需要的一些概念及基础知识，诸如图的概念与记号、图的着色问题、PCR技术等；第三部分给出本文的计算模型，包括算法步骤及其每步操作方法与技术；第四部分以一个61个顶点的图的3-着色为例，给出具体的生化实验方法步骤；第五部分是本章的理论分析部分，重点分析了模型中克服解空间指数爆炸问题的算法的复杂性问题；最后一部

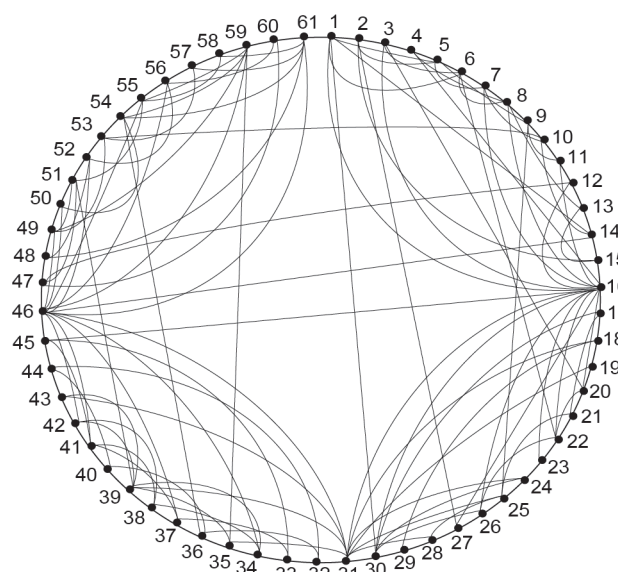


图1. 一个3-着色的61个顶点的图G。

分是文章的结论部分，对本文进行了总结，并指出进一步的研究方向。

2. 定义和符号

2.1. 图着色问题

本文所言之图皆指有限、无自环、无圈的无向简单图，通常用 G 表示图。我们用 $V(G)$ 、 $E(G)$ 分别表示图 G 的顶点集和边集。文中用 $\Gamma_G(v)$ 表示图 G 中顶点 v 的邻域，即在图 G 中与顶点 v 相邻的顶点构成的子集合。定义 $d_G(v)$ ，即顶点 v 在图 G 的度数。这些符号在不致混淆的情况下，分别简化为 V 、 E 、 d_G 和 Γ_G 。若用 $V = \{v_1, v_2, \dots, v_n\}$ 表示图 G 的顶点集合，则用 $d_i = d(v_i)$ 表示顶点 v_i 的度数，其中 $i = 1, 2, \dots, n$ 。图 G 中一条链是由顶点和边交替形成的序列，用 W 表示。链满足从一个顶点出发，到另一个顶点结束，且连续的顶点都是相邻的。 W 可以表述成：

$$W = v_1 \cdots v_k$$

式中， $k \geq 0$ ，并且对每个 $i = 1, 2, \dots, k-1$ ， v_i 和 v_{i+1} 是相邻的。链中边的数量即为链长。没有重复顶点的链构成路。

一个图 G 的着色是指对 G 中的每个顶点分配一种颜色，使得相邻的顶点着不同的颜色。换言之，是指对图 G 中的顶点集 $V(G)$ 的剖分：

$$V(G) = V_1 \cup V_2 \cup \cdots \cup V_k, \quad V_i \neq \phi (\text{空集}), \quad V_i \cap V_j \neq \phi, \quad i = 1, 2, \dots, k$$

满足图 G 正常着色的最小的颜色数称为色数，记为 $\chi(G)$ 。图 G 的一个 k -正常顶点着色，简称为图的 k -顶点着色，是指用 k [$k \geq \chi(G)$]种颜色对图 G 进行着色。通常用 $C_k(G)$ 表示图 G 的全体 k 着色构成的集合。由于本文仅考虑图的3-着色问题，故我们总是假定颜色集为 $C_3(G) = \{r, b, y\}$ ，其中， r 表示红色， b 表示蓝色， y 表示黄色。所以，求解图的3-着色问题可以认为是寻找从图的顶点集 $V(G)$ 到颜色集 $\{r, b, y\}$ 的一种映射： $f: V(G) \rightarrow \{r, b, y\}$ ，使得对于 $\forall uv \in E(G)$ ， $f(u) \neq f(v)$ 。图的3-着色问题是一个NP-完全问题[25]。文中的相关定义和符号参见文献[26]。

定义2.1 G 是一个简单图，顶点集合 $V(G) = \{v_1, v_2, \dots, v_n\}$ 。设 $v_{i1}v_{i2} \cdots v_{im}$ 是 $V(G)$ 中顶点的排序，其中 $v_{ij} \in V(G)$ ， $j = 1, 2, \dots, m$ ，且若 $j \neq l$ 则 $v_{ij} \neq v_{il}$ 。定义 $v_{i1}v_{i2} \cdots v_{im}$ 中的边数为 $N_{\text{edge}}(v_{i1}v_{i2} \cdots v_{im})$ ，满足：

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

2.2. PCR 技术

PCR法是一种可以快速并准确地大量复制DNA片段的技术。PCR的反应包括3个过程：变性、退火和延伸。当PCR产物的长度一定（相同）时，利用引物对序列作为目的序列，就可以得到首尾目的序列确定的所有的DNA片段，而中间未知的序列也将全部被扩增出来。PCR反应完成后，目标片段可以被大量地扩增出来，其数量可用公式 $y = (1 + x)^n$ 计算。其中， y 为DNA扩增倍数， x 为扩增效率， n 为循环数。

3. 模型及算法分析

本节将对1.2节中给出的计算模型的基本机理以及相关算法分析等给出较为详细的讨论。

3.1. 算法模型

在这个算法模型中，通过有限次的PCR操作，逐步删除不满足顶点着色的非解，扩增保留代表解的DNA分子。具体的算法流程图如图2所示。

3.2. 子图划分及桥点的确定

本模型计算时首先进行子图划分。子图划分的原因主要有如下两条：第一是易于生物操作实验；第二是为了较早地删除非解。这里 G_j ， $V(G_j) \subset V(G)$ ， $E(G_j) \subset E(G)$ ， $j = 1, 2, \dots, m-1$ 用来代表一级子图。子图划分方法步骤如下。

步骤1：确定子图的大小。一般划分的导出子图的顶点数以15~20为宜，原因是易于生物操作，并要求每个子图的规模基本接近，我们把这样划分的子图称为一级子图。划分前需要分析整体图，要求：每个子图的边数尽可能得多，因为边数越多，在一级子图处理中就可以尽可能多地删除非解。由于在一级子图中删除非解生化操作方便，且在一级子图中删除的非解越多，后面合成的二级子图中的非解自然越少，越容易进行相应的生化操作。例如，对于如图3中所示的图给出了两种不同的划分，一种是： G'_1 ，以 $V'_1 = \{1, 2, 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 22\}$ 为顶点子集，和 G'_2 ，以 $V'_2 = \{8, 9, 10, 11, 12, 13, 14, 15, 23, 24, 25, 26, 27, 28, 29, 30\}$ 为顶点子集；另外一种划分是： G'_3 ，以 $V'_3 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ 为顶点子集，和 G'_4 ，以 $V'_4 = \{16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30\}$ 为顶点子集。显然，后一种划分比前一种划分好。

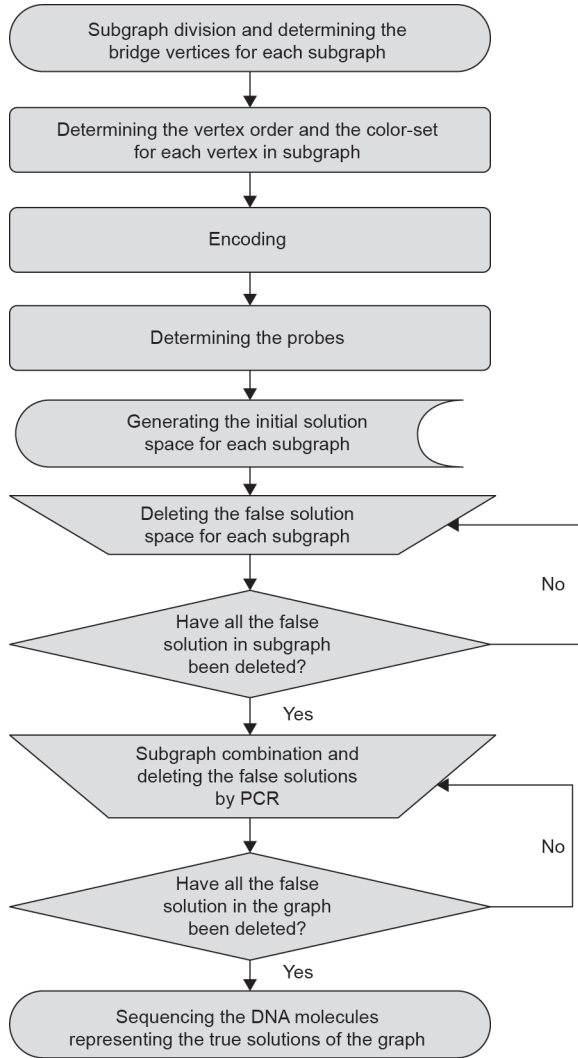


图2. 模型的算法流程图。

步骤2: 确定第一个子图, 并确定该子图的两个桥点 u_1 和 u_2 。这两个桥点必须相邻, 且在该子图中度数之和最大; 若满足该条件的顶点对不止一对, 则选择以此两点为起点与终点的顶点排序后边数最多的那一对, 即满足 $N_{\text{edge}}(u_1 = v_{i_1}v_{i_2}\cdots v_{i_m} = u_2)$ 最大 (详见3.3节); 若满足上述三个条件的不止一对, 则选择其中之一即可。

步骤3: 确定第二个子图 G_2 , 该子图包含 u_2 。在子图 G_2 中找一个新的桥点 u_3 , 寻找的方法是: 与 u_2 相邻的且在 G_2 中度数最大的。选择条件同步骤2。

步骤4: 重复步骤3, 直到完成最后一个子图的划分。

值得注意的是, 第一, 最后一个子图的顶点数目一般与前面子图的顶点数不同; 第二, 最后一个子图 G_m 的桥点有可能是前一个子图的一个桥点 u_1 与第一个子图的桥点 u_{m-1} 构成。

桥点是本模型中的关键顶点, 这是因为是通过桥点

利用PCR技术很容易将各种子图之间合并起来。我们在子图桥点的确定时要求: 在一个子图中两个桥点的度序列之和尽可能得大。这是因为, 两个顶点度序列之和越大, 即与这两顶点相邻的顶点越多, 因此, 相应顶点的颜色集合中颜色数必然减少越多, 从而使得初始解空间变得越小。但对于所有顶点度数均相等的情况, 我们所选择的桥点应遵循3.3节中关于子图顶点排序与每个顶点颜色集的确定。图4给出了图1中所示图 G 被划分的4个子图, 其中顶点1,16,31,46,61是5个桥点。

3.3. 子图顶点排序与子图中每个顶点颜色集的确定

完成子图划分后, 每个子图的桥点也随之确定。本小节所言的子图顶点排序就是求解出该子图以两个桥点为起点和终点的一种顶点排序, 使其该排序中相邻顶点之间的边数 (见定义2.1) 尽可能得多。以子图 G_1 为例, 若子图 G_1 的顶点集为 $V(G_1) = \{v_1, v_2, \dots, v_t\}$, 令 v_1 、 v_t 是它的两个桥点, 不失一般性, 设排序的顶点序列为 $v_1 = v_{i_1}v_{i_2}v_{i_3}\cdots v_{i_t} = v_t$, 需要满足:

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

为了给出下面顶点排序算法, 我们在此先引入根点路集的概念。设 v 是图 G 的一个顶点。我们把以顶点 v 为起点的 G 中所有路径构成的路径集称为 v -根点路径集, 或简称为 v -路径集, 并记作 $P(v)$ 。求一个顶点的路径集是不困难的, 如首先取 v 的邻域 $\Gamma(v)$, 接着再对 $\Gamma(v)$ 中每个顶点求邻域, 记作 $\Gamma^2(v)$, $\Gamma^2(v)$ 中不含元素 v , 如此下去, 得到诸如 $\Gamma^3(v)$ 、 $\Gamma^4(v)$ 等, 将在 $|V(G)|-1$ 步之内终止, 这种方法实际上是所谓的剪枝算法, 是一种NP-算法, 但文中所确定的图的子图的顶点数在20之内, 故易于实现。

设 G_1 一个划分子图。 v_1 和 v_t 是它的两个桥点, 则子图 G_1 顶点排序的算法步骤如下。

步骤1: 求出两个桥点 v_1 和 v_t 的根点路径集 $P(v_1)$ 和 $P(v_t)$, 若在 $P(v_1)$ 中含有子图 G_1 的Hamilton路, 则选择该路径, 否则, 转向步骤2。

步骤2: 若 $P(v_1) \cup P(v_t) = V(G_1)$, 则从 $P(v_1)$ 与 $P(v_t)$ 中分别选择出满足下式的两条路径 $P_1 = v_{i_1}v_{i_2}v_{i_3}\cdots v_{i_r}$, $P_t = v_{i_s}v_{i_{s+1}}\cdots v_{i_t}$; $\max|E(P_1)|+|E(P_t)|$; $P_1 \in P(v_1)$, $P_t \in P(v_t)$, $V(P_1) \cap V(P_t) = \phi$ 。

步骤3: 若 $V' = V(G_1) - P(v_1) \cup P(v_t) \neq \phi$ 。则从导出子图 $G[V']$ 中找出最长的路径, 记作 $P' = u_1u_2\cdots u_m$ 。若 $V'' = V(G_1) - V(P_1) - V(P_t) - V' = \phi$, 则子图 G_1 的顶点排序为: $v_{i_1}v_{i_2}v_{i_3}\cdots v_{i_r}u_1u_2\cdots u_mv_{i_s}v_{i_{s+1}}\cdots v_{i_t}$ 。若 $V''' = V(G_1) - V(P_1)$

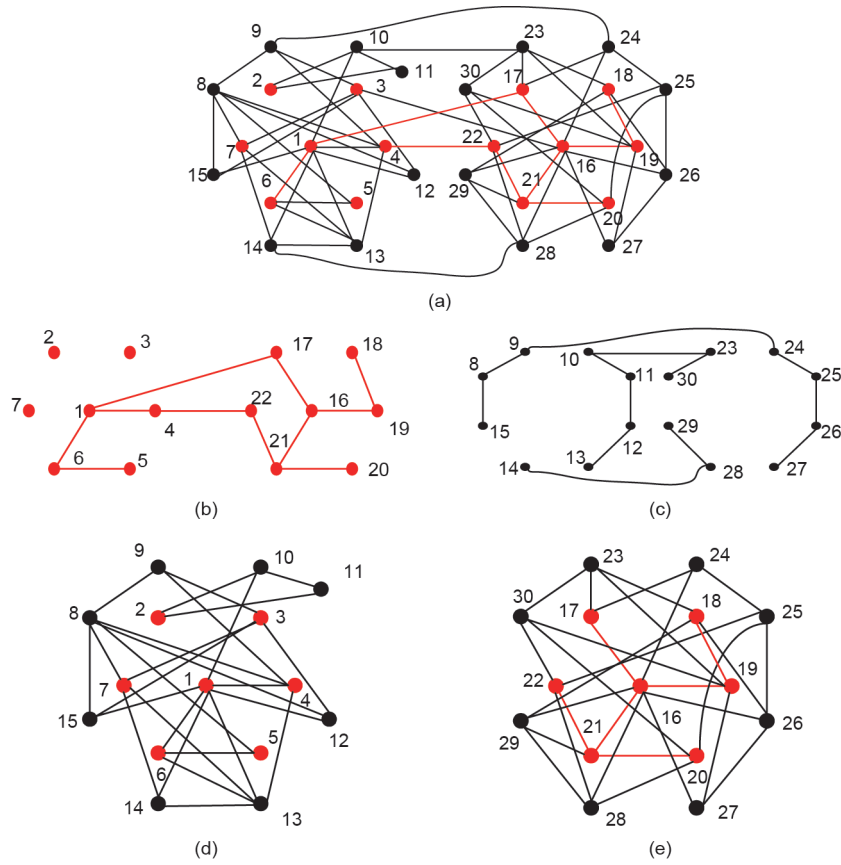


图3. 一个一级子图划分的例子。(a) 图G; (b) 子图 G_1' ; (c) 子图 G_2' ; (d) 子图 G_3' ; (e) 子图 G_4' 。

– $V(P_i) - V' = \{u_{i1}u_{i2} \cdots u_{iq}\} \neq \phi$, 则子图 G_1 的顶点排序为:
 $v_1v_2v_3 \cdots v_{i1}u_1u_2 \cdots u_m u_{i1}u_{i2} \cdots u_{iq}v_{i3}v_{i(s+1)} \cdots v_t$ 。

对于图3中所示的子图 G_3' , 它的两个桥点为1与13, 我们在桥点1的根点路径集中找到路径1-10-11-2; 桥点13的根点路径集中找到与之相匹配的路径13-14-7-8-15-3-9-4, 在剩余的顶点中得到5-6,12, 因此, 顶点排序为: 1,10,11,2,5,6,12,4,9,3,15,8,7,14,13 (图5)。

在图4中第一个子图中顶点中顶点的排序是: 1,2,⋯,16, 这种排序是最好的, 每对相邻的顶点 i 和 $i+1$ 均有边相邻, $i = 1, 2, \cdots, 15$ 。但若将它们次序排成1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 1, 则只有5条边: $\{5, 7\}$, $\{9, 11\}$, $\{13, 15\}$, $\{15, 2\}$, $\{6, 8\}$ 。

步骤4: 顶点重新排序。当子图的顶点排序按着步骤1的方法确定之后, 然后对图的顶点进行重新标定。假定标定前的子图的顶点排序为 $v_1v_2 \cdots v_t$, 顶点重新排列后的序列为 $v_{\sigma(1)}v_{\sigma(2)} \cdots v_{\sigma(t)}$, 显然, 此排序就是顶点下标通过了下列置换而获得:

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

如对图2中子图 G_3' 的一种顶点排序 (见图5) 后, 顶

点重新排序的映射为

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

不失一般性, 我们顶点排序后的子图顶点进行重新标定, 标定的原则是: 令顶点 $\sigma(i)$ 标定为 i , 其中 $i = 1, 2, \cdots, t$ 。

我们在本文中的颜色数取 $k = 3$, 并且颜色集为 $C(3) = \{\text{red}, \text{yellow}, \text{blue}\} = \{r, y, b\}$, 用 r_i, y_i, b_i 分别表示图中的顶点 v_i 着红色、黄色和蓝色。这种确定颜色集合的方法可以推广到颜色数 $k > 3$ 的情况。

由于每个子图的首尾顶点 (桥点) 相邻, 不失一般性, 对于第一个子图 G_1 , 它的两个桥点分别是 v_1 和 v_t 。令顶点 v_1 着红色, 记作 r_1 , 顶点 v_t 着蓝色, 记作 b_t (注: 顶点 v_t 也可以着黄色, 我们在此先不予考虑, 按照蓝色的情况下计算出所有解后, 只需要对颜色集做一个颜色置换即可。即在红色不变的情况下, 将黄色与蓝色互换即可得到另外一种着色, 对于其他的子图同样, 今后不再说明)。除顶点 v_t 外, 顶点 v_1 的领域 $\Gamma(v_1)$ 中的每个顶点着蓝色或黄色两种; 同理, 除顶点 v_1 外, 顶点 v_t 的领域 $\Gamma(v_t)$ 中的顶点着红色或黄色两种。对于图1中所示的4个

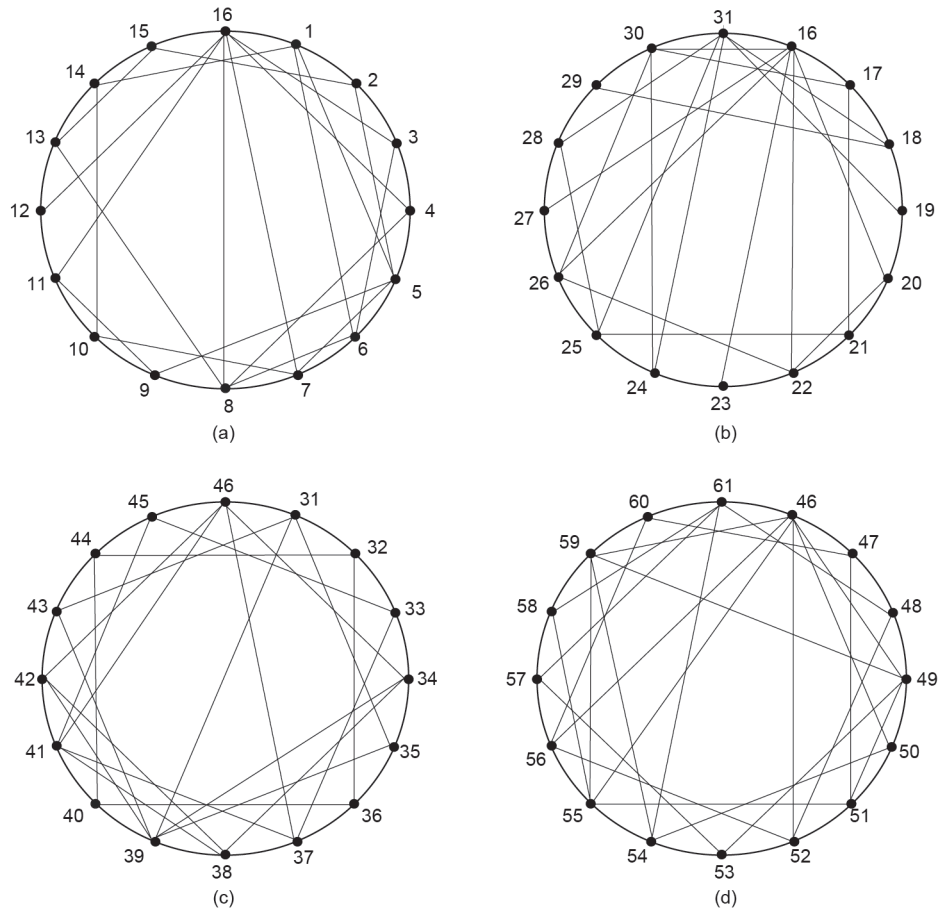


图4. 图1中图 G 的一级子图划分的4个子图。(a) 子图 G_1 ; (b) 子图 G_2 ; (c) 子图 G_3 ; (d) 子图 G_4 。

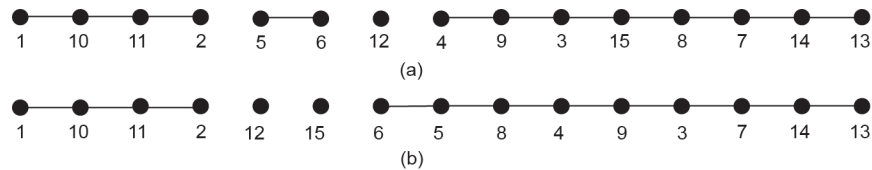


图5. 图3中子图 G_3 的一种顶点排序及对应的11条边的集合。

子图的颜色集合表分别如图6所示。此方法可使得初始解空间复杂度得到大大降低，有关详细讨论见5.2节。

3.4. DNA 序列的编码

由于DNA计算中作为“数据”的DNA序列在“运算过程”中是通过特异性杂交来实现的，因此，DNA计算中的编码要受到诸如解链温度、DNA序列的相似度、 ΔG 等众多条件的约束，而且要受到诸如问题的规模、链的长度、实验环境等诸多因素的影响。因此，DNA计算中的编码问题是一个非常复杂而困难的问题。许多学者视DNA序列中的编码问题为NP-完全问题，目前已有不少学者在不同约束条件下关于编码问题的算法。本文采用的编码不仅考虑特异性杂交反应，而且要考虑重点

采用PCR技术的编码，综合设计，给出如下约束条件：

- (1) 所有编码没有超过连续4个A、4个T、4个C或4个G的现象；
- (2) 任意编码的GC含量为40%~60%；
- (3) 任意两条编码没有连续8个以上的碱基是相同的；
- (4) 任意一条编码自身不存在连续4个碱基的互补序列；
- (5) 任意一条编码的3'或5'末端的连续5个碱基和其他编码中的任意5个碱基不能相同；
- (6) 作为引物的DNA序列自身形成二聚体的化学自由能变化的绝对值 ($\Delta|G|$) 一般不超过 $6.0 \text{ kcal} \cdot \text{mol}^{-1}$ ($1 \text{ cal} = 4.1868 \text{ J}$)；

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r_1	r_3	r_4				r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}		r_{15}	
	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	
	b_2			b_5	b_6			b_9	b_{10}			b_{13}	b_{14}		b_{16}
(a)															
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	r_{17}			r_{20}	r_{21}	r_{22}	r_{23}			r_{26}	r_{27}		r_{29}		r_{31}
	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}	y_{22}	y_{23}	y_{24}	y_{25}	y_{26}	y_{27}	y_{28}	y_{29}	y_{30}	
b_{16}		b_{18}	b_{19}		b_{21}			b_{24}	b_{25}			b_{28}	b_{29}		b_{30}
(b)															
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
r_{31}		r_{33}	r_{34}		r_{36}	r_{37}	r_{38}		r_{40}	r_{41}	r_{42}		r_{44}	r_{45}	
	y_{32}	y_{33}	y_{34}	y_{35}	y_{36}	y_{37}	y_{38}	y_{39}	y_{40}	y_{41}	y_{42}	y_{43}	y_{44}	y_{45}	
	b_{32}	b_{33}		b_{35}	b_{36}			b_{38}	b_{39}	b_{40}			b_{43}	b_{44}	b_{46}
(c)															
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
	r_{47}	r_{48}	r_{49}	r_{50}	r_{51}	r_{52}	r_{53}	r_{54}	r_{55}	r_{56}	r_{57}	r_{58}	r_{59}	r_{60}	
	y_{47}		y_{49}	y_{50}	y_{51}	y_{52}	y_{53}			y_{55}	y_{56}		y_{59}		y_{61}
b_{46}		b_{48}			b_{51}		b_{53}	b_{54}				b_{57}	b_{58}		b_{60}
(d)															

图6. 确定图4中4个子图的颜色集。其中设定 $C(1)=\{r_1\}$, $C(16)=\{b_{16}\}$, $C(31)=\{r_{31}\}$, $C(46)=\{b_{46}\}$ 以及 $C(61)=\{y_{61}\}$ 。(a)子图 G_1 ; (b)子图 G_2 ; (c)子图 G_3 ; (d)子图 G_4 。

(7) 作为引物的DNA序列间形成的二聚体的 $|\Delta G|$ 一般不超过 $9.0 \text{ kcal}\cdot\text{mol}^{-1}$, 而引物间3'端形成的二聚体的 $|\Delta G|$ 则不能高于 $6.0 \text{ kcal}\cdot\text{mol}^{-1}$ 。

根据以上约束条件, 搜索计算所需要每个代表顶点颜色的DNA序列。我们在搜索计算时采用了文献[27]中的方法进行了初步搜索计算, 然后在得到的DNA序列中选择出所需要的DNA序列。选择原则是: 若搜索的DNA序列的长度小于15碱基, 从长度为20碱基中选择; 若搜索的DNA序列的长度大于20碱基, 则选择颜色数达到要求的碱基长度最小的DNA序列, 并进而采用最新的Preimer软件(5.0)进行挑选检测。

定理3.1: 令代表顶点 v_i 可能的颜色的DNA序列标记为 x_i , $x \in \{r, b, y\}$, $i = 1, 2, \dots, n$, 它的Watson-Crick互补的序列则用 \bar{x}_i 表示。根据每个顶点可能的颜色集, 可以确定, 对于一个任意的图来讲, 所需要编码的DNA序列的数量为

$$N_{\text{DNA}} \leq 3n - d(v_1) - d(v_n) - 2(3 - 1) \quad (3)$$

本文需要代表颜色的129个DNA序列如表1所示。

3.5. 探针的确定

DNA计算机的优点是巨大的并行性、海量的信息存储能力等。通过十多年的研究, 我们以为DNA计算机的最大优势应该是利用合成仪器合成的DNA序列仍具有一定的生物活性, 其中氢键之间的引力是最为主要

的。利用这一特性, 可以将大量的非解排除在初始解空间之外, 这一优势, 电子计算机是无法比拟的。而实现这一思想的方法就是如何设计杂交中的探针。本模型关于探针的设计方法步骤如下(以颜色数为 $k = 3$ 为例说明)。

步骤1: 按照上节的方法确定给定图 G 的颜色集: 令 $V(G) = \{v_1, v_2, \dots, v_n\}$, 并令按照1的顶点排序为 v_1, v_2, \dots, v_n , 定义 C_{v_i} 表示顶点 v_i 的可着颜色集, $i = 1, 2, \dots, n$ 。

步骤2: 探针集的确定。探针只在由步骤1中确定排序相邻的两个顶点之间建立, 即只在顶点 v_i 与 v_{i+1} 之间建立, 其中 $i = 1, 2, \dots, n-1$ 。这里将探针记作 $\overline{x_i x_{i+1}}$ 。若 v_i 与 v_{i+1} 在 G 中相邻, 则 x_i 和 x_{i+1} 不能取相同颜色; 若 v_i 与 v_{i+1} 在 G 中不相邻, 则 x_i 和 x_{i+1} 可以取相同颜色。例如, 图4中给出的子图 G_1 中顶点7的颜色集合为 $\{r_7, y_7\}$, 顶点8的颜色集合为 $\{r_8, y_8\}$, 由图3可知, 这两个顶点在子图 G_1 中相邻, 故只有相异颜色之间有探针: $\overline{r_7 y_8}$ 和 $\overline{y_7 r_8}$ 。

步骤3: 每个探针的DNA序列构成: 对探针 $\overline{x_i x_{i+1}}$, 用代表 x_i 的DNA序列的后半段和 x_{i+1} 的前半段组成的序列的补链构成。 $y_7 = 5'-\text{AATACGCACTCATCA CATCG}-3'$, $r_8 = 5'-\text{GACCTTACCGTTTAGAGTCG}-3'$, 则有 $y_7 r_8 = 5'-\text{CATCACATCGGACCTTACCG}-3'$, $\overline{y_7 r_8} = 5'-\text{CGGTAAGGTCCGATGTGATG}-3'$ 。

为了更清晰地反映探针集合, 我们在此引入图论方法给予描述, 并且这种方法在后面关于降低初始解空间分析方面很有用。

一个图 G 的探针图, 记作 $B(G)$, 在不致混淆的情况下简记为 B , 定义为

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

依据探针的定义, 在图3与图4的基础上, 我们在图7中分别给出了图4中所示4个子图3-着色的探针图, 分别记为 B_1 、 B_2 、 B_3 和 B_4 。从每个探针图中, 很清楚地看到该图的探针数目及分布情况。

3.6. 初始解空间的合成

采用Adleman[2]的方法进行初始解空间合成。将代表各个顶点可能的颜色的DNA序列 x_{v_i} 和各种探针 $\overline{x_{v_i} x_{v_{i+1}}}$ 混合后, 通过T4连接酶作用后, 产生新的长DNA序列则代表每个子图可能解的初始解空间。

表1 代表每个顶点的可能着色的DNA序列

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 CATTG-3'	y_6	5'-CGTGA TTGTT GGAAT 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 TTTCG-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 GCGA 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 TCGCA CAAGC-3'	b_{25}	5'-GATTA TCGTC CAGCC TTCTC-3'
r_{26}	5'-CCTCC GTAGT TATTG ATGCC-3'	y_{26}	5'-GTTAC GGTG 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'-TAAAGT 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 TTAAT CGTCG TTCGC-3'	r_{31}	5'-GCTCA GGAGT GTTA 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 CTAAT 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 TTAAT TCAGG-3'	y_{44}	5'-CAAAG AGTTA GTCGG GTCTG-3'

x_i	DNA sequence	x_i	DNA sequence
b_{44}	5'-CTTCT ATGTT TAGCC CGAGG-3'	r_{45}	5'-GCAGG ACAAG GCTCA TAGTT-3'
y_{45}	5'-GTGAC GCCAT CATTG GAGAT-3'	b_{46}	5'-CTATC AGAAA CCGGT CAGAG-3'
r_{47}	5'-GCTGA CTCA 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 TGA CT 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 TTAG-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'		

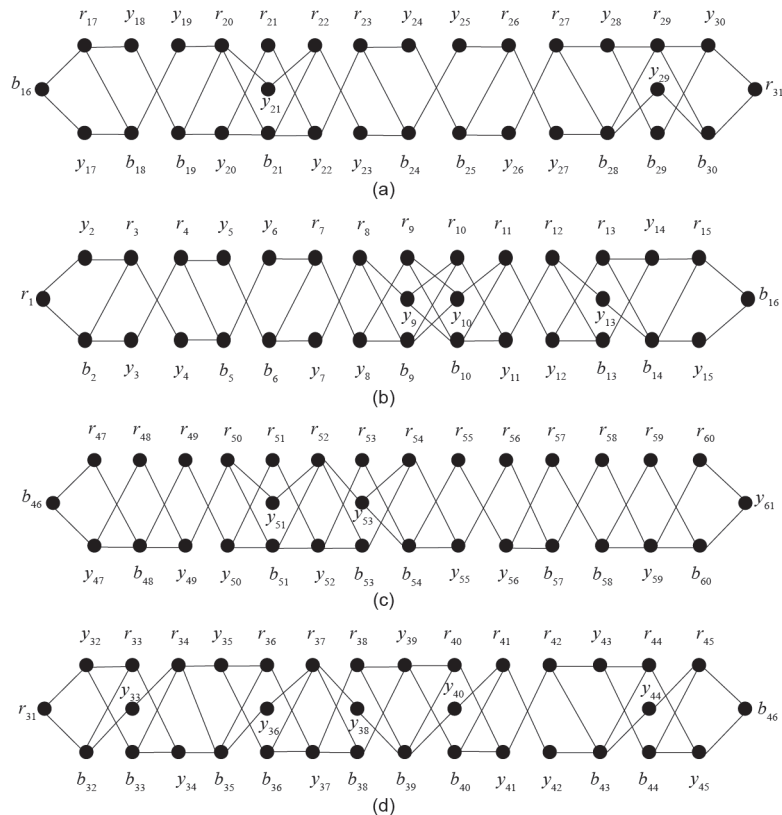


图7. 图4中4个子图相应的探针图。(a) B_1 ; (b) B_2 ; (c) B_3 ; (d) B_4 。

这里用 $S(G_j)$ 代表生成的初始解空间，即集合 $S(G_j) = (x_{v_{j1}} x_{v_{j2}} \cdots x_{v_{jt}})$ ，这里 $x \in \{r, b, y\}$ ， $j=1, 2, \dots, m$ 。如果每个 $x_{v_{jt}}$ 长为 l 个碱基，且每个子图的顶点个数为 t ，则所生成的

代表可能解的DNA序列长为 $t \times l$ 个碱基， $j=1, 2, \dots, m$ 。

这里要说明的是，在合成每个子图初始解空间的时候，如果顶点 v_i 和 v_{i+1} 之间有边，则在杂交反应中，采用

集合 P_1 中的探针，由于这些探针本身就不包含两个顶点着相同颜色的情况，则在初始解空间时，就可以删除一些相邻顶点着同色的非解。因此，在构建初始解空间时，若子图中含有的边 $e = \{v_i, v_{i+1}\}$ 的数量比较多，则可以删除较多的非解。

对于每个子图的合成初始解空间的具体步骤为：先用T4多核苷酸激酶（T4 PNK）对DNA序列 x_{v_i} 进行5'端磷酸化。其次，将磷酸化产物与探针在 $10 \times$ T4PNK缓冲液中进行退火，反应条件为： 94°C 、5 min、 50°C 、10 min。再次，使用T4连接酶（T4 DNA ligase）对退火产物进行连接反应， 16°C 过夜。最后以连接产物为模板，以 $\langle x_{v_{j_1}}, \overline{x_{v_{j_2}}} \rangle$ ， $x \in \{r, b, y\}$ ， $j = \{1, 2, \dots, m\}$ 为引物对，进行PCR扩增，得到的DNA序列的集合，即为每个子图的初始解空间。

3.7. 非解删除

本小节给出从3.5节中所建立的每个子图初始解空间中删除非解的方法与步骤，为方便起见，我们在此先引入一些基本记号。令 $E_j^1 = \{v_i v_{i+1} \in E(G_j), v_i v_i \in E(G_j); i = 1, 2, \dots, t-1\}$ ， $E_j^2 = \{v_1 v_i \in E(G_j); i = 2, 3, \dots, t\}$ ， $E_j^3 = E(G_j) - E_j^1 - E_j^2$ ， $j = 1, 2, \dots, m$ 。

删除非解实际上就是将 E_j^3 中的每条边两端具有同色的DNA序列从初始解空间 $S(G_j)$ 中删除，主要是利用PCR技术来删除非解。在具体的删除非解的过程中，我们将 E_j^3 中的边分为两种情况，对不同的情况，具体操作分正向边和单边两种情况。

3.7.1. 正向边

在由 E_j^3 导出子图中，首先选择较长的路或者圈来从 $S(G_j)$ 中删除非解。文中建立所谓的并行性PCR方法来删除非解。令 G_1 为含有顶点子集 $V(G_1) = \{v_1, v_2, \dots, v_t\}$ 的子图， v_1, v_2, \dots, v_t 为 G_1 的顶点序列（排序方法见3.3节）。 $\forall e = v_i v_j \in E(G_1)$ ，若 $i < j$ ，则 $e = v_i v_j$ 是一个正向边；否则即为反向边。如果路中所有的边均为正向边，则该路为正向路，否则为反向路。

其具体方法在此以路 $P = v_1 v_j v_y v_z$ ， $1 \leq i \leq j \leq y \leq z \leq t$ ，为例给予说明（其他的路类似进行）。这条路上其实包含三条边： $v_1 v_j$ 、 $v_j v_y$ 和 $v_y v_z$ ，对此情况，在第一次PCR时，分别以 $\langle x_1, \overline{x_i} \rangle$ 、 $\langle x_i, \overline{x_j} \rangle$ 、 $\langle x_j, \overline{x_y} \rangle$ 、 $\langle x_y, \overline{x_z} \rangle$ 和 $\langle x_z, \overline{x_t} \rangle$ 为引物对进行扩增。此时 x_i 和 x_j 、 x_j 和 x_y 、 x_y 和 x_z 均不能同时取同色，这样DNA序列就被分成5个片段，然后逐级合并成长为 $t \times l$ 个碱基的DNA序列。利用此方法，能同时对多条边所对应的非解都进行

删除。具体实现步骤见4.4节，理论分析5.3节。这样就可以使得利用较少次的PCR反应，同时删除多条边对应的非解。

3.7.2. 单边

在对 E_j^3 导出子图中所有路或圈删除非解后，有可能剩下一些单边，它们所对应的非解删除方法如下：对一条单边 $v_i v_j$ 删除非解，需要进行3次PCR反应，即可将这条边对应的非解删除。令 $x_i \in C_{v_i}$ ， $x_j \in C_{v_j}$ ，且在第一次PCR时，将分别以 $\langle x_1, \overline{x_i} \rangle$ 、 $\langle x_i, \overline{x_j} \rangle$ 和 $\langle x_j, \overline{x_t} \rangle$ 为引物进行扩增，这里， x_i 与 x_j 不能同时取同色，这样代表顶点 v_i 和 v_j 着同色的DNA链将不被扩增出来，使得非解被删除。当完成了对每个子图的 E_j^3 中的边删除非解后，就可得到代表每个子图 G_j （ $j = 1, 2, \dots, m$ ），正常着色的DNA序列。

3.8. 子图合并及新子图中的非解删除

在获得了代表每个子图的解的DNA链之后，通过对子图进行合并，并删除合并后代表非解的DNA链，主要包括三步。

步骤1：合并子图 G_j 和 G_{j+1} （ $j = \{1, 2, \dots, m\}$ ）。利用桥点的DNA序列，以PCR为工具，将删除非解后的代表这两个子图的正常着色的DNA序列拼接在一起。若 $|V(G_j)| = t_1$ ， $|V(G_{j+1})| = t_2$ ，则形成长为 $(t_1 + t_2 - 1) \times l$ 个碱基的DNA序列，这些DNA序列将作为图 G 的二级导出子图 $G[V(G_j) \cup V(G_{j+1})]$ 的所有可能的解。

步骤2：按照3.7节所述的非解删除方法，对每个导出子图 $G[V(G_j) \cup V(G_{j+1})]$ 进行删除非解，更确切地讲，删除边集 $E_{j,j+1} = \{uv \in E(G); u \in V(G_j), v \in V(G_{j+1})\}$ 中每条边所对应的非解，即可得到代表其正常着色的DNA序列。

步骤3：对二级或更高级的导出子图重复步骤1和步骤2的操作，直至合并为图 G ，并得到代表图 G 正常着色的DNA序列。

3.9. 解的检测

采用测序的方法得到最终的每个子图的解的DNA序列，以及代表图 G 的解的DNA序列，并利用软件Gene Tool进行分析，得到图 G 的正常着色方案。

4. 实例与实现

为了实现所设计的图顶点着色DNA计算模型，本节

对图1中对61个顶点图 G 的3-着色问题进行生化试验验证。这个实验规模不仅是迄今为止DNA计算研究中实验规模最大的一个实验，而且因为其通用性，证明了它的搜索计算能力可以超越 $O(3^{59})$ 。

4.1. 子图划分及颜色集确定

由3.2节子图划分的方法以及子图顶点排序的方法，我们最终得到图 G 的4个子图如图4所示。按照3.3节的方法，4个子图的颜色集为图6所示。

4.2. 编码

按照3.5节中给出的编码方法，代表每个顶点可能的颜色的DNA序列如表1所示。根据3.5节中探针的构建方法和每个子图所对应的探针图（图7），共需要构建的探针的数目为185条： $\overline{r_1y_2}, \overline{r_1b_2}, \overline{y_2r_3}, \overline{b_2r_3}, \overline{b_2r_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}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_{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}}, \overline{r_{60}y_{61}}, \overline{b_{60}y_{61}}。$

所有的129条DNA序列和185条探针序列均由生工生物工程（上海）股份有限公司合成。

4.3. 构建初始解空间

对每个子图采用了PCR技术合成初始解空间。具体以第一子图为例说明：将代表颜色的33条DNA序列的5'端磷酸化后，与相应的46条探针混合，用T4DNA连接酶进行连接后，再用引物对 $\langle \overline{r_1}, \overline{b_{16}} \rangle$ 进行扩增，得

到大小为320 bp的DNA序列集合即为初始解空间，见图8 (a)。第二、三和四子图初始解空间的构建与第一个子图相同，见图8 (b)~(d)。采用Gel Extract on Kit (U-gene)，将回收产物溶于50 μL 的无菌水中，测其浓度为400 $\text{ng}\cdot\mu\text{L}^{-1}$ 。

对各子图的初始解空间进行检测。以第一子图为例，分别用 $\langle x_1, \overline{x_i} \rangle$ ，其中， $x = \{r, b, y\}$ ， $i = 2, 3, \dots, 16$ 为引物，初始解空间的DNA序列稀释50倍后作为模板，进行PCR反应，判断初始解空间是完备的。PCR的反应体系和反应条件同上。这里以子图 G_1 的检测结果为例 [图8 (e)、(f)]，其余相同。

4.4. 子图删除非解

将这些边分成可以形成路或圈以及边，分别针对三个基本类型给出实验结果。

4.4.1. 路和圈

对于，路和圈采用了所谓的并行性PCR方法来删除非解。即由每次删除一条边，变成了同时删除多条边，使得PCR操作次数大大减少。在此以子图 G_2 中的路 $P = 16-20-22-26-30$ 为例给出实验结果：

首先，分别以 $\langle \overline{b_{16}}, \overline{r_{20}} \rangle$ 、 $\langle \overline{b_{16}}, \overline{y_{20}} \rangle$ 、 $\langle \overline{r_{20}}, \overline{y_{22}} \rangle$ 、 $\langle \overline{y_{20}}, \overline{r_{22}} \rangle$ 、 $\langle \overline{r_{22}}, \overline{y_{26}} \rangle$ 、 $\langle \overline{y_{22}}, \overline{r_{26}} \rangle$ 、 $\langle \overline{r_{26}}, \overline{y_{30}} \rangle$ 、 $\langle \overline{r_{26}}, \overline{b_{30}} \rangle$ 、 $\langle \overline{y_{26}}, \overline{b_{30}} \rangle$ 、 $\langle \overline{y_{30}}, \overline{r_{31}} \rangle$ 和 $\langle \overline{b_{30}}, \overline{r_{31}} \rangle$ 为引物对，以初始解空间中DNA链稀释50倍为模板，把全长为320 bp的第二子图库链DNA序列分解成了几个大小不同的片段：100 bp (16~20)、60 bp (20~22)、100 bp (22~26)、100 bp (26~30)和40 bp (30~31) [图9 (a)、(b)]。其中，16~20指从顶点16到顶点20，其余类同。图9 (a)中泳道5的引物对为 $\langle \overline{r_{22}}, \overline{y_{26}} \rangle$ ，无电泳条带产生，是由于在构建初始解空间时不存在同时含有 r_{22} 与 y_{26} 的DNA序列。

其次，进行该圈第二次PCR操作。分别以 $\langle \overline{b_{16}}, \overline{y_{22}} \rangle$ 为引物，以图9 (a)泳道1和3的PCR产物的混合物为模板，得到了140 bp的电泳条带；以 $\langle \overline{r_{26}}, \overline{r_{31}} \rangle$ 为引物，以图9 (b)泳道1和4的PCR反应产物的混合物为模板，得到了120 bp的电泳条带；以 $\langle \overline{r_{26}}, \overline{r_{31}} \rangle$ 为引物，以图9 (b)泳道2和5的PCR反应产物的混合物为模板，得到了120 bp的电泳条带。电泳结果见图9 (c)。

再次，进行该圈第三次PCR操作。以 $\langle \overline{b_{16}}, \overline{r_{26}} \rangle$ 为引物，以图9 (a)泳道6和图9 (c)泳道1的PCR产物的混合物为模板，得到了220 bp的电泳条带。电泳结果见图9 (d)。

最后，进行该圈第四次PCR操作。以 $\langle \overline{b_{16}}, \overline{r_{31}} \rangle$ 为引

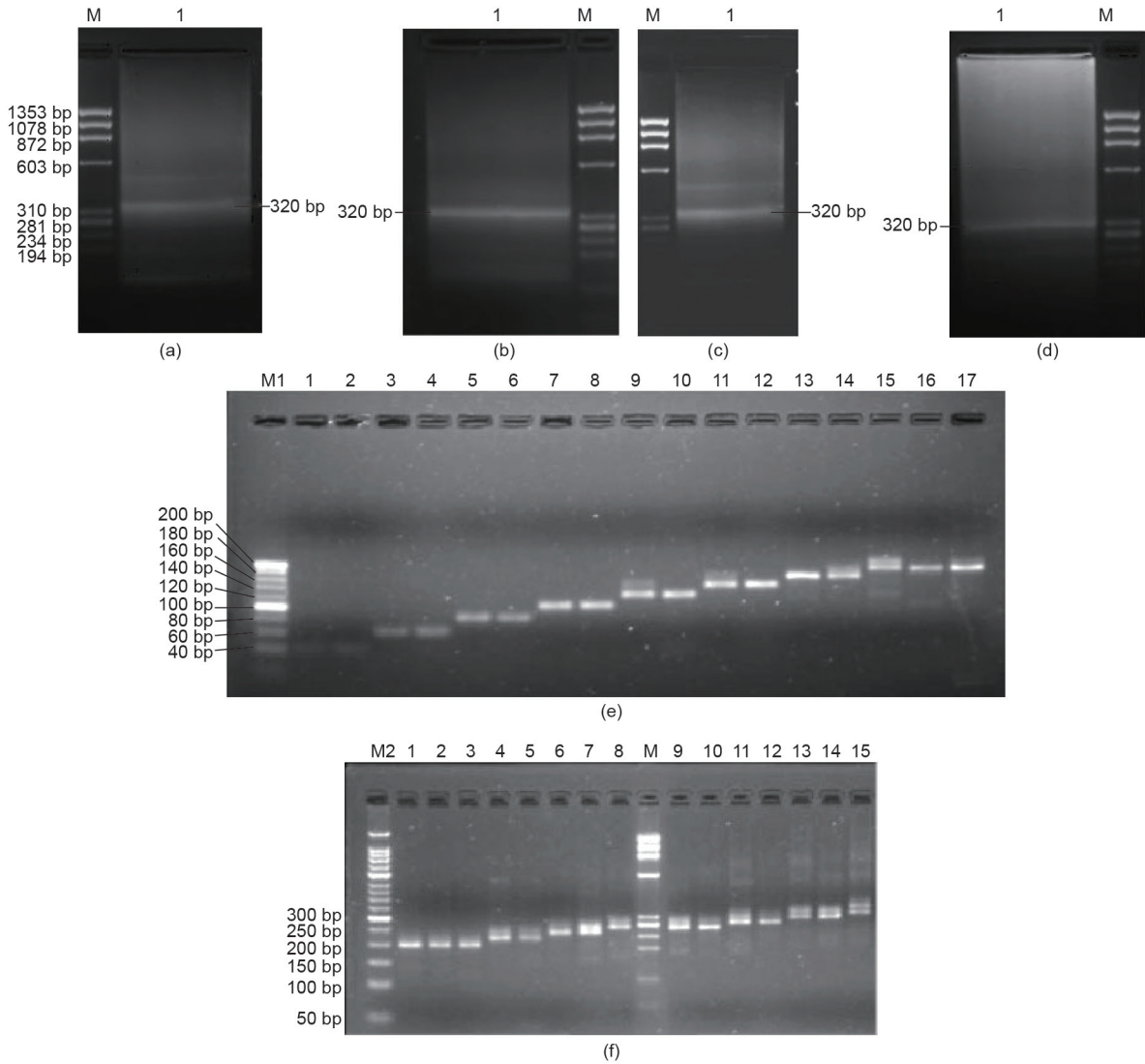


图8. 初始解空间的构建及检测, 图中M即为 $\phi 174$ -Hae III digest DNA marker. (a)~(d) 中的泳道1为所构建的每个子图的初始解空间, 其中, (a)~(d) 分别对应 $S(G_1)$ 、 $S(G_2)$ 、 $S(G_3)$ 和 $S(G_4)$, 每个DNA序列的大小为320 bp. (e) 中1~17 对应的引物对为 $\langle r_1, \bar{y}_2 \rangle$ 、 $\langle r_1, \bar{b}_2 \rangle$ 、 $\langle r_1, \bar{r}_3 \rangle$ 、 $\langle r_1, \bar{y}_3 \rangle$ 、 $\langle r_1, \bar{r}_4 \rangle$ 、 $\langle r_1, \bar{y}_4 \rangle$ 、 $\langle r_1, \bar{y}_5 \rangle$ 、 $\langle r_1, \bar{b}_5 \rangle$ 、 $\langle r_1, \bar{y}_6 \rangle$ 、 $\langle r_1, \bar{b}_6 \rangle$ 、 $\langle r_1, \bar{r}_7 \rangle$ 、 $\langle r_1, \bar{y}_7 \rangle$ 、 $\langle r_1, \bar{r}_8 \rangle$ 、 $\langle r_1, \bar{y}_8 \rangle$ 、 $\langle r_1, \bar{r}_9 \rangle$ 和 $\langle r_1, \bar{b}_9 \rangle$. (f) 中1~15泳道对应的引物对为 $\langle r_1, \bar{r}_{10} \rangle$ 、 $\langle r_1, \bar{y}_{10} \rangle$ 、 $\langle r_1, \bar{b}_{10} \rangle$ 、 $\langle r_1, \bar{r}_{11} \rangle$ 、 $\langle r_1, \bar{y}_{11} \rangle$ 、 $\langle r_1, \bar{r}_{12} \rangle$ 、 $\langle r_1, \bar{y}_{12} \rangle$ 、 $\langle r_1, \bar{r}_{13} \rangle$ 、 $\langle r_1, \bar{y}_{13} \rangle$ 、 $\langle r_1, \bar{b}_{13} \rangle$ 、 $\langle r_1, \bar{y}_{14} \rangle$ 、 $\langle r_1, \bar{b}_{14} \rangle$ 、 $\langle r_1, \bar{r}_{15} \rangle$ 、 $\langle r_1, \bar{y}_{15} \rangle$ 和 $\langle r_1, \bar{b}_{16} \rangle$.

物, 分别以图9 (d) 泳道1和图9 (c) 泳道2的PCR产物的混合物为模板; 以图9 (d) 泳道1和图9 (c) 泳道3的PCR产物的混合物为模板进行, 分别得到了两个全长320 bp的电泳条带, 电泳结果见图9 (e)。此步骤的具体操作示意图如图9 (f) 所示。上述反应结束后, 我们可以发现, 生成了两组解: $b_{16}r_{20}y_{22}r_{26}y_{30}r_{31}$ 和 $b_{16}r_{20}y_{22}r_{26}b_{30}r_{31}$ 。这两组解将作为新的模板, 进入下一次非解删除过程。

4.4.2. 两端颜色未定的边

对于两端颜色未定的边, 采用了PCR技术删除非解。在此以第一个子图中的边 $e = \{4,8\}$ 为例给出实验

结果。

首先, 以上轮PCR的产物为模板, 分别以 $\langle r_1, \bar{r}_4 \rangle$ 、 $\langle r_4, \bar{y}_8 \rangle$ 、 $\langle y_8, \bar{b}_{16} \rangle$ 、 $\langle r_1, \bar{y}_4 \rangle$ 、 $\langle y_4, \bar{r}_8 \rangle$ 和 $\langle r_8, \bar{b}_{16} \rangle$ 为引物对进行第一次PCR, 结果如图10 (a) 所示。其中, 当用 $\langle y_4, \bar{r}_8 \rangle$ 为引物时, 没有PCR产物生成。

其次, 以 $\langle r_1, \bar{y}_8 \rangle$ 为引物, 以图10 (a) 的泳道1和3 PCR产物的混合物为模板, 进行第二次PCR操作, 得到了160 bp的电泳条带, 见图10 (b)。

最后, 以 $\langle r_1, \bar{b}_{16} \rangle$ 为引物, 以图10 (a) 泳道5和图10 (b) 泳道1的 PCR产物的混合物为模板进行PCR反应, 得到了全长为320 bp的电泳条带[图10 (c)]。至此 $e = \{4,8\}$ 的三次PCR操作完成, 得到全长为320 bp的DNA序

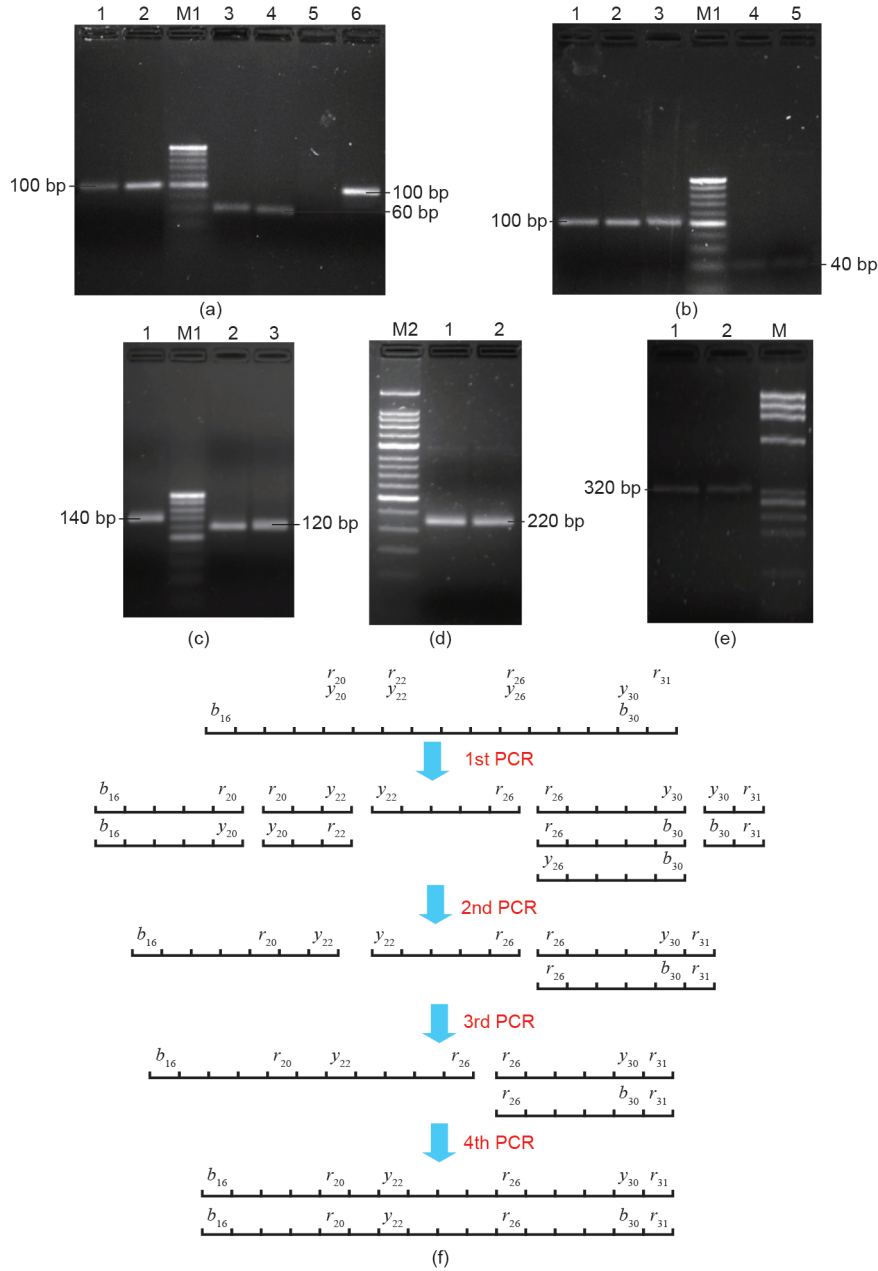


图9. 对圈 $C = 16-17 \cdots 31-16$ 删除非解的实验结果。(a) 泳道1~6对应的引物对为 $\langle b_{16}, \overline{r_{20}} \rangle$ 、 $\langle b_{16}, \overline{y_{20}} \rangle$ 、 $\langle r_{20}, \overline{y_{22}} \rangle$ 、 $\langle y_{20}, \overline{r_{22}} \rangle$ 、 $\langle r_{22}, \overline{y_{26}} \rangle$ 和 $\langle y_{22}, \overline{r_{26}} \rangle$; (b) 泳道1~5对应的引物对为 $\langle r_{26}, \overline{y_{30}} \rangle$ 、 $\langle r_{26}, \overline{b_{30}} \rangle$ 、 $\langle y_{26}, \overline{b_{30}} \rangle$ 、 $\langle y_{30}, \overline{r_{31}} \rangle$ 和 $\langle b_{30}, \overline{r_{31}} \rangle$; (c) 泳道1~3对应的引物对为 $\langle b_{16}, \overline{y_{22}} \rangle$ 、 $\langle r_{26}, \overline{r_{31}} \rangle$ 和 $\langle r_{26}, \overline{r_{31}} \rangle$; (d) 泳道1~2对应的引物对为 $\langle b_{16}, \overline{r_{26}} \rangle$; (e) 泳道1~2对应的引物对为 $\langle b_{16}, \overline{r_{31}} \rangle$; (f) PCR操作流程示意图。

列的集合，且DNA序列中代表顶点4和8的着色方案已经确定为 r_4 和 y_8 。

4.4.3. 相关联的顶点的颜色已确定的边

对于相关联的顶点的颜色已确定的边，利用了PCR技术进行检测，删除非解。以第三子图的边 $e = \{(34), (39)\}$ 为例进行给出实验结果，如图11所示。当在子图 G_3 中完成删除圈 $C = 34-38-42-46-34$ 的所有PCR反

应，以及完成了圈 $C = 34-38-42-46-34$ 第一次PCR反应之后，形成了代表6条解的DNA序列，其中顶点39和42分别为 b_{39} 、 y_{39} 、 r_{42} 和 y_{42} 。为了删除非解，利用引物对 $\langle y_{39}, \overline{y_{42}} \rangle$ ，以6条DNA序列为模板，进行PCR反应，此时，出现DNA片段的则代表非解[图11(a)]。在图11(a)中，泳道5、6、7、8、11和12中出现了长为80 bp的电泳条带。实验结果表明，编号为(3)、(4)和(6)的DNA序列将被删除。实验方案示意图如图11(b)所示。

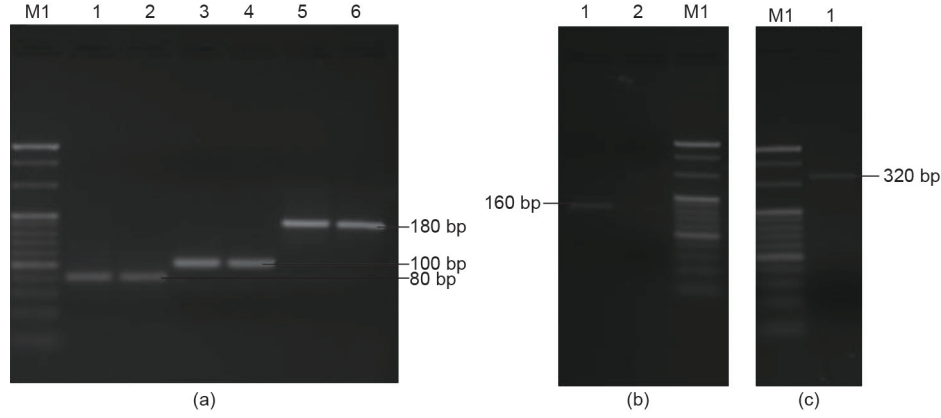


图10. 对边 $e = \{4,8\}$ 删除非解的实验结果。(a) 泳道1和2、3和4、5和6的PCR产物分别以 $\langle y_8, \overline{b_{16}} \rangle$ 、 $\langle r_4, \overline{y_8} \rangle$ 和 $\langle r_1, \overline{r_4} \rangle$ 为引物对；(b) 中泳道1的PCR产物以 $\langle r_1, \overline{y_8} \rangle$ 为引物对；(c) 中泳道1的PCR产物相同，以 $\langle r_1, \overline{b_{16}} \rangle$ 为引物对。

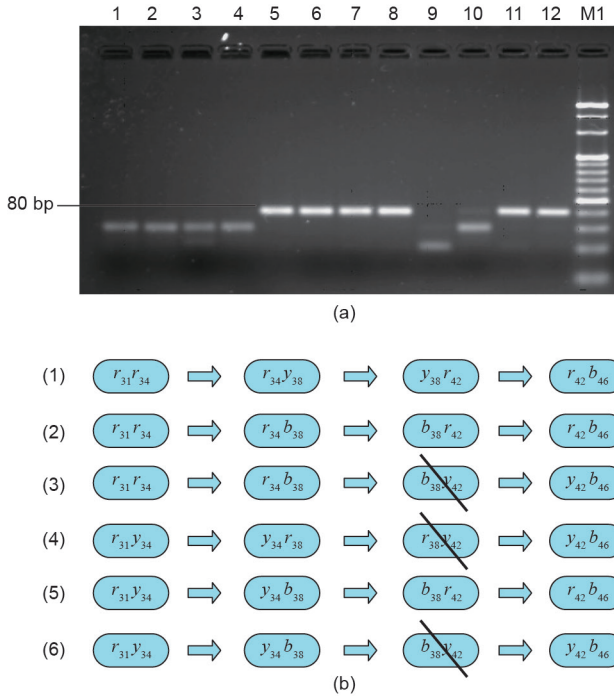


图11. 对边 $e = \{(34), (39)\}$ 删除非解的实验结果。(a) 所有泳道对应的引物对均为 $\langle y_{39}, \overline{y_{42}} \rangle$ ，泳道1和2的模板对应解(1)，泳道3和4的模板对应解(2)，泳道5和6的模板对应解(3)，泳道7和8的模板对应解(4)，泳道9和10的模板对应解(5)，泳道11和12的模板对应解(6)；(b) 非解删除示意图：“\”代表操作后被删除的非解。

4.5. 子图合并及删除非解

完成了对每个子图的并行操作后，需要找出代表图1中所示的图 G 的正常3-着色的DNA链。这就需要将各个子图的解合并，并进一步删除非解(见4.4节)。具体合并和删除非解的结果见下文。

第一步将经过上述反应后，将代表满足子图 G_1 和 G_2 正常3-着色的解的长度为320 bp的DNA序列的混合物两两混合作为模板，以 r_1 与 $\overline{r_{31}}$ 为引物，进行PCR扩增，得到长度为620 bp的DNA链，就是图 G 中导出的

子图 $G[V_1 \cup V_2]$ 的所有可能的3-着色解，也就是子图 $G[V_1 \cup V_2]$ 的初始解空间。然后对此初始解空间利用与上述子图非解的删除方法完全来删除其中的非解。得到的结果记为 $X_i, i = 1, 2, 3, 4$ ，见图12 (a)、(b)。

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

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

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

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

同时用相同的方法处理子图 G_3 和 G_4 ，得到的结果记为 $Y_i, i = 1, 2, 3, 4, 5, 6$ ，见图12 (c) ~ (e)。

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

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

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

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

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

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

将得到全长为620 bp的DNA序列混合作为模板，以 $\langle r_1, \overline{y_{61}} \rangle$ 为引物对，进行PCR扩增，得到长度为1220 bp的DNA序列集合，其中也含有图 G 非解。用同样的方法

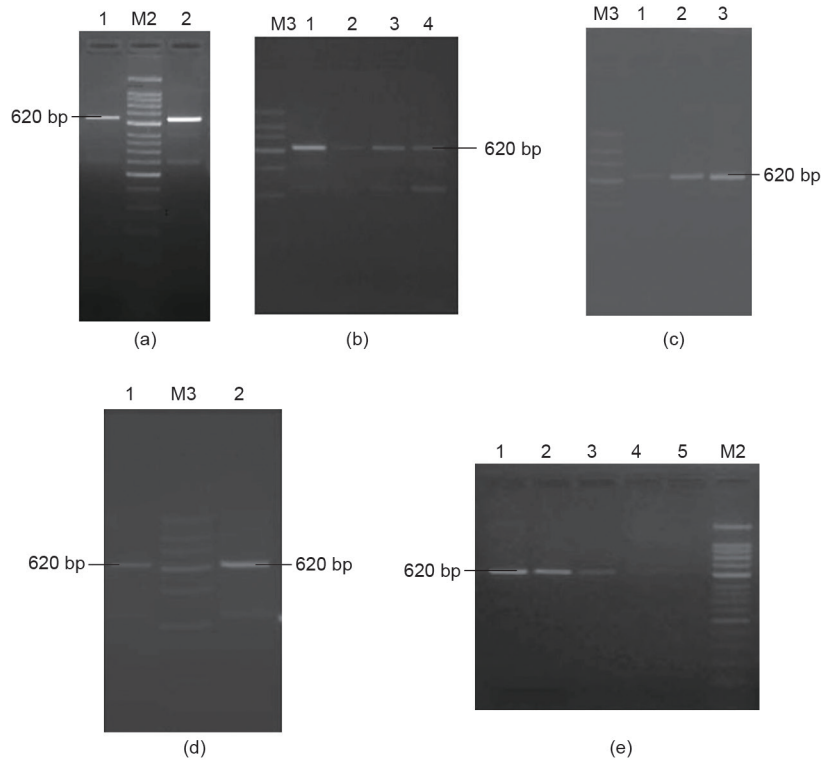


图12. 子图 $G[V_1 \cup V_2]$ 和 $G[V_3 \cup V_4]$ 的解。(a) 泳道1和2对应解 X_1 和 X_2 ; (b) 泳道1和2对应解 X_3 , 泳道3和4对应解 X_4 ; (c) 泳道1和2对应解 Y_1 , 泳道3对应解 Y_2 ; (d) 泳道1对应解 Y_3 , 泳道2对应解 Y_4 ; (e) 泳道1对应解 Y_5 , 泳道2对应解 Y_6 。

进行删除非解, 最后得到的DNA序列即代表了满足图 G 正常3-着色的解。共有8条DNA序列, 分别代表8个解(图13)。

将图13中的8条DNA序列回收后, 连接在pMD 19-T载体上, 并转化 $E. coli$ DH5 α 后, 在北京奥科生物技术有限公司进行测序, 得到满足图 G 正常3-着色的解, 记为 $Z_i (i = 1, 2, \dots, 8)$ 。

$$\begin{aligned}
 Z_1 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} \\
 &\quad y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} r_{60} y_{61} \\
 Z_2 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} \\
 &\quad y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} b_{60} y_{61} \\
 Z_3 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} \\
 &\quad y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} r_{60} y_{61} \\
 Z_4 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} y_{18} b_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38}
 \end{aligned}$$

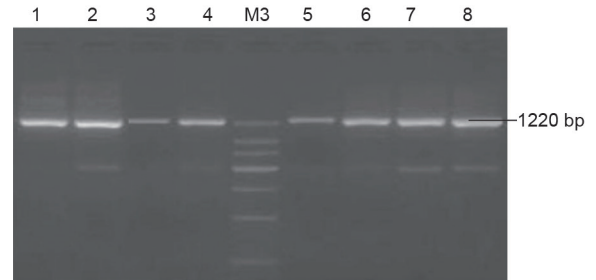


图13. 图 G 的解, M3为150 bp ladder的DNA marker。泳道1~8分别对应解 $Z_1 \sim Z_8$ 。

$$\begin{aligned}
 &\quad y_{39} b_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} b_{60} y_{61} \\
 Z_5 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} \\
 &\quad y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} r_{60} y_{61} \\
 Z_6 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38} \\
 &\quad y_{39} r_{40} y_{41} r_{42} b_{43} y_{44} r_{45} b_{46} y_{47} b_{48} r_{49} y_{50} b_{51} r_{52} y_{53} b_{54} r_{55} y_{56} \\
 &\quad r_{57} b_{58} y_{59} b_{60} y_{61} \\
 Z_7 &= r_1 b_2 y_3 r_4 y_5 b_6 r_7 y_8 r_9 b_{10} y_{11} r_{12} b_{13} y_{14} r_{15} b_{16} r_{17} b_{18} y_{19} r_{20} \\
 &\quad b_{21} y_{22} r_{23} b_{24} y_{25} r_{26} y_{27} b_{28} r_{29} y_{30} r_{31} b_{32} y_{33} r_{34} b_{35} y_{36} r_{37} b_{38}
 \end{aligned}$$

$$\begin{aligned}
& y_{39}b_{40}y_{41}r_{42}b_{43}y_{44}r_{45}b_{46}y_{47}b_{48}r_{49}y_{50}b_{51}r_{52}y_{53}b_{54}r_{55}y_{56} \\
& r_{57}b_{58}y_{59}r_{60}y_{61} \\
Z_8 = & r_1b_2y_3r_4y_5b_6r_7y_8r_9b_{10}y_{11}r_{12}b_{13}y_{14}r_{15}b_{16}r_{17}b_{18}y_{19}r_{20} \\
& b_{21}y_{22}r_{23}b_{24}y_{25}r_{26}y_{27}b_{28}r_{29}y_{30}r_{31}b_{32}y_{33}r_{34}b_{35}y_{36}r_{37}b_{38} \\
& y_{39}b_{40}y_{41}r_{42}b_{43}y_{44}r_{45}b_{46}y_{47}b_{48}r_{49}y_{50}b_{51}r_{52}y_{53}b_{54}r_{55}y_{56} \\
& r_{57}b_{58}y_{59}b_{60}y_{61}
\end{aligned}$$

5. 复杂性分析

本节在理论上较为系统地分析了文中所提出模型的算法复杂性问题，重点集中在降低初始解空间与为提高生物操作速度而提出的并行PCR操作方法。在初始解空间的复杂性方面，给出了一般计算公式，并分析了最坏与最好的情形。为了分析方便，在5.1节中引入图中链的计算公式，该公式可用于计算图中路的计数研究。

5.1. 图中链的计数公式

本节给出求解一个给出图的任意两个顶点之间的walk(链)数目的计算公式，这个公式可用于路径的计数，进而用于分析通过合理地减少顶点颜色降低初始解空间复杂度的大小。

引理5.1[28] 设 G 是一个 n 子图，且 $V(G) = \{v_1, v_2, \dots, v_n\}$ 。 A 是 G 的相邻矩阵。那么从顶点 v_i 开始到 v_j 结束的长为 l 的路径的数目为矩阵 A^l 中 (i, j) 位置的值。这里 A^l 是 l 个 A 相乘后得到的矩阵。

5.2. 降低初始解空间复杂性的分析

在过去已有的一些用于NP-完全问题的DNA计算模型中，以为DNA分子是纳米级的，可以用大量的DNA分子来表示无穷的数据。而一般很少考虑在初始解空间中删除大量非解的优化计算的思想，总是采用枚举方法。事实证明这种用空间换时间的思想是不可取的。原因有两个：第一，随着问题规模的增大，所需要编码的DNA数量随之增大，就有可能无法得到足够数量的DNA分子；第二，即使通过增大DNA分子的长度，找到了一定数量的DNA分子，但由于DNA分子的增长，则会导致后续的生化试验操作的难度大大增加，不仅影响实验结果的可靠性，甚至无法操作。因此，如何降低DNA计算中初始解空间的DNA分子数量是至关重要的。

为了降低求解图顶点着色问题DNA计算模型的初始解空间，模型中采用了两种方法：第一种方法是3.2节中所给出的子图划分方法；第二种方法是3.3节给出的子图顶点排序方法。并在这两种方法的基础上给出子图中每个顶点颜色集。在子图颜色集的基础上，引入了探针图的概念。

实际上，我们引入探针图已经将顶点排序与减少顶点颜色集这两种方法的信息全部包括在其中。下面，我们利用探针图来给出模型中初始解空间中不同DNA序列数目的计算公式。这个公式由下述定理给出。

定理5.2 设 G 是一个 n 子图， G_1 是按照3.2节中给出的一个子图。若 $V(G_1) = \{v_1, v_2, \dots, v_t\}$ ，且顶点1与顶点 t 分别是它的两个桥点。令 $B(G_1)$ 表示 G_1 的探针图，则 G_1 的初始解空间中不同DNA链的数目是图 $B(G_1)$ 中从顶点1到顶点 t 的长度为 $t-1$ 的路径的数目，记作 $N_p(1, t)$ ，由如下公式给出：

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

证明 由探针图 $B(G_1)$ 的定义知， $B(G_1)$ 中的顶点 x_i 代表图中顶点 v_i 可着的颜色， $B(G_1)$ 的边 $\{x_i, z_{i+1}\}$ 表示 x_i 与 z_{i+1} 之间构成的探针，即表示在初始解空间中，当顶点 v_i 着色为 x 时，顶点 v_{i+1} 可以着颜色 z ， $x, z \in \{r, y, b\}$ 。因此， $B(G_1)$ 中任意一条从顶点 r_1 到顶点 b_t 的路径代表了子图 G_1 的一种着色，而从顶点 r_1 到顶点 b_t 的所有路径代表了初始解空间中全体可能的解。

进一步，我们来考察 $B(G_1)$ 图的结构，该图可分为 t 组，下标为 i 的表示第 i 组。由 $B(G_1)$ 的定义知道：第 i 组只与第 $i+1$ 组之间有边相连，且一定有边相连。因此，在探针图 $B(G_1)$ 中，从顶点 r_1 到顶点 b_t 的每条路径的长度只能为 $t-1$ ，且从顶点 r_1 到顶点 b_t 的每条walk就是从顶点 r_1 到顶点 b_t 的每条路径。故由引理5.1，本定理获证。证毕。

推论5.3 图4所示的子图 G_1 、 G_2 、 G_3 和 G_4 中初始解空间中不同DNA链的数目分别是：89、81、412和151条；删除枚举型初始解空间中非解比例是：99.9998%、99.9998%、99.999%和99.9997%。

证明 由定理5.2知，图 G_1 中初始解空间中DNA链的数目就是 G_1 的探针图 B_1 (图7)中从顶点 r_1 到顶点 b_{16} 之间长度为15的数目 $N_{33}(r_1, b_{16})$ 。

为了计算 $N_{33}(r_1, b_{16})$ ，利用引理5.1，先给出图 B_1 的相邻矩阵：

桥点，即或 $v_{i0} = v_1, v_{im} \neq v_t$ ；或 $v_{i0} \neq v_1, v_{im} = v_t$ ；对此情况，需要增加一个桥点 v_i 或 v_t ，如图14 (b)所示。

情况3：路径的两个端点 v_1 和 v_t 都是子图的桥点，即 $v_{i1} = v_1$ 且 $v_{it} = v_t$ ，对此情况，只考虑路径 P 的情况，如图14 (c)所示。

由于桥点的颜色已经确定，故上述情况2与情况3中与桥点相邻的顶点的着色集合中不含桥点的颜色。基于并行PCR技术中顶点的确定（见上述三种情况），我们统一成上述情况1的情况，即对于情况2和3中删除与桥点相邻的边：从情况2中的路径 P 删除一条边 $\{v_{i1}, v_{i2}\}$ 或 $\{v_{i(m-1)}, v_{im}\}$ ，得到与情况1类似的路径 P' ；从情况3中的路径 P 中删除两条边 $\{v_{i1}, v_{i2}\}$ 和 $\{v_{i(m-1)}, v_{im}\}$ ，得到与情况1类似的路径 P' 。

下面来介绍并行PCR方法，仅考虑上述情况1的路径。首先按照图14中的方法确定正向路径 $P = v_{i0}v_{i2}\dots v_{im}$ 的顶点数目及次序；然后进行PCR操作。其实，PCR扩增的次数只与由图14中确定的顶点数或者边数有关。如对于情况1，第一次PCR扩增时的扩增对分别为： $\langle v_1, v_{i0} \rangle, \langle v_{i0}, v_{i1} \rangle, \dots, \langle v_{i(m-1)}, v_{im} \rangle, \langle v_{im}, v_t \rangle$ ；第二次PCR是在第一次PCR结果的基础上，分别从两边以相邻的两个片段进行PCR扩增；第三次在第二次的基础上再从两边分别以相邻的两个片段进行PCR扩增。在图15中，我们分别以5个顶点（即两条边路径）和6个顶点（即3条边的路径）的情况给予说明，这里的连接边表示对两个或多个顶点之间实行的PCR扩增。

由图15可以看出，边数是2的路径需要进行PCR的次数为3，边数为6时需要PCR的次数为4。一般的，边数为 x 时，按着并行PCR方法，需要多少次？我们有如下定理。

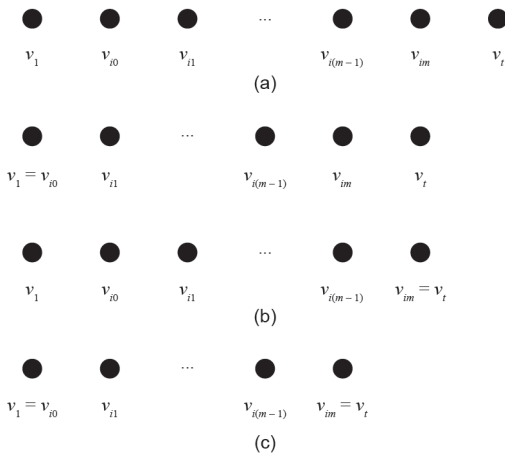


图14. 正向路径与桥点的三种情况。

定理5.4 设 P 是子图 G_1 中一条边数为 x 的正向路径。当 $2^{l-1} < x \leq 2^l$ 时，定义 $\langle x \rangle = 2^l$ ，则并行PCR操作次数，记作 $PCR(x)$ ，应为

$$PCR(x) = 1 + \log_2 \langle x + 2 \rangle, \quad x \geq 1 \quad (6)$$

证明 根据对情况1进行证明。此时，路径的两个端点 v_1 和 v_t 不是子图的桥点，即 $v_{i0} \neq v_1, v_{im} \neq v_t$ ；如果路径 P 包含 x 条边，那么路径的顶点数目为 $x+1$ ，由于此路径不含桥点 v_1 和 v_t ，按照并行PCR方法，需要再增加两个桥点，使得顶点数为 $x+3$ [图14 (a)]。假设 $x + 2 = 2^l$ ，第一次PCR操作的顶点对的数目即为增加了桥点 v_1 和 v_t 之后的边数，为 $(x + 3) - 1 = x + 2 = 2^l$ 。按照并行PCR的定义，第二次操作顶点对的数目为 $(x+2)/2, 2^{l-1}$ 个，第3次PCR操作对的数目为 $2^{l-2}, \dots$ ，第 l 次PCR操作对的数目为 $2^{l-(l-1)} = 2$ ，第 $l+1$ 次操作即可完成全部PCR操作，即推出：当 $x + 2 = 2^l$ 时，PCR操作的次数 $l+1$ 。所以 $PCR(x) = l + 1 = 1 + \log_2 2^l = 1 + \log_2 \langle 2^l \rangle = 1 + \log_2 \langle x + 2 \rangle$ 。即证明了当 $x + 2 = 2^l$ 时结论成立。同理可以证明当 $x + 2 = 2^{l+1}$ 时结论成立，即 $PCR(x) = l + 2$ 。

下面，我们来证明 $x + 2 = 2^l + 1$ 的情况。

类似于上述证明，路径 P 的顶点数为 $x + 1$ ，加上两个桥点，使得顶点数为 $x + 3$ ，当 $x + 2 = 2^l + 1$ 时，即 $x + 3 = 2^l + 2$ 个顶点，因此，第一次PCR操作对的数目为 $(x + 3) - 1 = x + 2 = 2^l + 1$ ；第二次操作对的数目为 $2^{l-1} + 1$ 个，第3次PCR操作对的数目为 $2^{l-2} + 1, \dots$ ，第 l 次PCR操作对的数目为 $2^{l-(l-1)} + 1 = 3$ ，第 $l + 1$ 次操作的次数为2，第 $l + 2$ 即可完成全部PCR操作。另一方面，将 $x + 2 = 2^l + 1$ 代

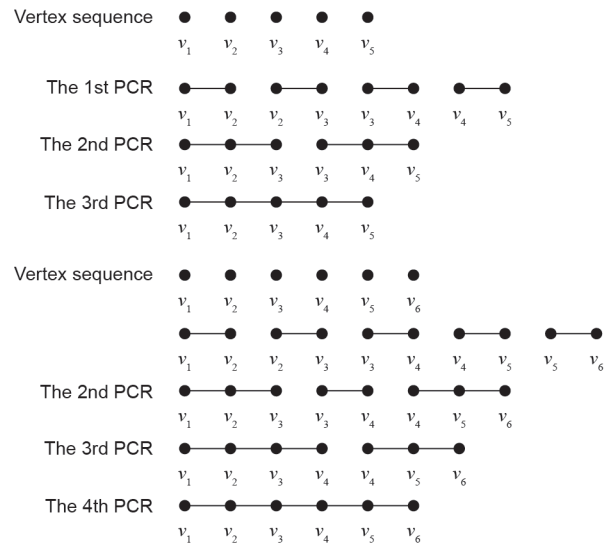


图15. 并行PCR扩增技术示意图。

表2 逐边删解与并行PCR方法PCR操作次数对照表

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.

入式 (5.1), 有 $\text{PCR}(x) = l + 2 = 1 + (l + 1) = 1 + \log_2 2^{l+1} = 1 + \log_2 \langle 2^l + 1 \rangle = 1 + \log_2 \langle x + 2 \rangle$, 即证明了当 $x + 2 = 2^{l+1}$ 时结论成立。

由于对于任意 $2 \leq y < 2^l$, $x + 2 = 2^l + y$ 对应的PCR操作次数不小于 $x + 2 = 2^l + 1$ 对应的PCR操作, 但不大于 $x + 2 = 2^{l+1}$ 对应的PCR操作次数, 而这两个PCR操作次数均为 $l + 2$, 这就证明了, 对于任意 $2 \leq y < 2^l$, $x + 2 = 2^l + y$ 时, 结论成立。正如上面所述, 图14中第二种情况和第三种情况最终都可以转换成第一种情况, 所以本定理得证。

定理5.4完整地刻画了并行PCR操作时, 其正向路径的长度 x 与PCR操作次数的关系。显然, 并行PCR操作可大大降低操作次数, 使得DNA计算机的运行速度得到极大的提高。我们知道, 逐边删除非解, 即一条一条地删除非解, 每次需要3次PCR(见第4节)。下面, 我们通过表2与图16中曲线可以清楚地看到并行PCR的优势。

进一步, 我们有:

推论5.5 设 P 是子图 G_1 中边数为 x 的一个正向路径, 则有如下结论:

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

证明 令 $x + 2 = 2^l + y$, $1 \leq y < 2^l$, 则 $\langle x + 2 \rangle = 2^{l+1} = 2(2^l + y - y) < 2(2^l + y)$; 又因为 $2^l > y$, 所以, $2^l + 2^l > y + 2^l$, 即 $2^{l+1} > y + 2^l$, 从而有:

$y + 2^l < 2^{l+1} < 2(2^l + y)$, 此即 $x + 2 < \langle x + 2 \rangle < 2(x + 2)$, 从而有 $\log_2(x + 2) < \log_2 \langle x + 2 \rangle < \log_2 2(x + 2)$, 因而,

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

由洛必达法则有

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

从而本推论获得证明。

推论5.5说明, 随着子图中正向路径 P 中边数的增加, 并行PCR方法操作的优势越大, 即DNA计算的速

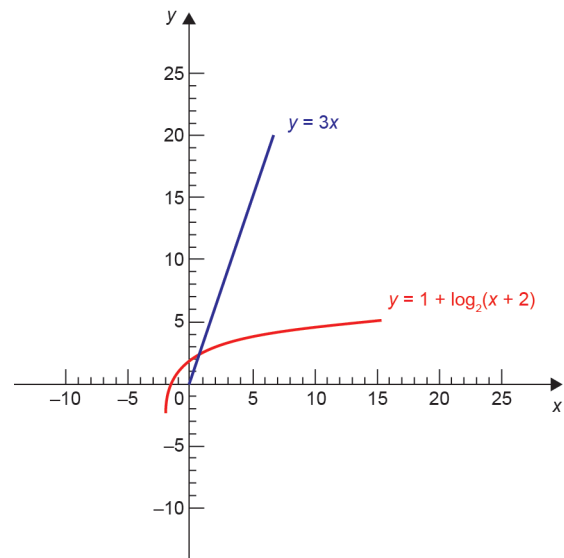


图16. 逐边删解与并行PCR方法PCR操作次数曲线。其中, x 表示正向路径 P 的边数, y 表示PCR操作次数。

度越快。由于每次PCR需要的时间较长(约半个小时), 因此, 更显示出并行PCR方法的优势。

6. 结论与讨论

本文建立了一种用于求解图顶点着色的DNA计算模型。本文对该模型进行了详细的论述。该模型延续了上一章中介绍的DNA计算模型的优势, 同时利用对图进行分解与合并的操作, 使得在实际处理中, 大部分的生化实验可以并行操作, 进而使得该模型可用于求解较大规模的图顶点着色问题。本文利用该模型成功地对一个含有61个顶点的图进行了求解。

在目前已建立的DNA计算模型中, 几乎都采用枚举方法来构建初始解空间。而用这种方法构建的初始解空间中自然存在大量非解, 因此也就不能避免解空间指数爆炸问题。而且, 利用这种方法, 还可以使编码的数量多, 长度增加, 进而加大了生化操作的难度和实验费用。本模型所建立的子图初始解空间提出了两个原则: 一个是图顶点序列的首、尾点的度数尽可能大; 另一个是子

图的顶点序列中相邻的顶点在原图中尽可能有边相连。按照这两个原则,该DNA计算模型在构建初始解空间的时候,就删除了相当数量的非解,使得初始解空间不再枚举,进而回避了初始解空间指数爆炸问题。通过计算,我们对每个子图所建立非枚举型的初始解空间中不同DNA链的数目分别为89、81、304和151,极大地降低了初始解空间,不但回避了解空间指数爆炸问题,而且可以使得生物操作更加方便,生物结果更加可靠。

通过子图的分解与合并,可以同时处理多个子图。在文章中,我们同时对四个子图进行了处理,并且同时对图 G 导出子图 $G[V_1 \cup V_2]$ 和 $G[V_1 \cup V_2]$ 也采用并行处理。而且,子图分解后,解空间也随之减少。从而使得一个复杂的NP-完全问题在一个较短的时间内更加容易地完成了。

并行操作这也是本模型的一个关键技术创新点,它使得实验速度得到了极大的提高。这里的平行操作主要包括两点:第一,因为子图的分解而导致的并行操作;第二,实验中对图的路或者圈的删除非解的操作也是并行的。在实验过程中,我们将许多条边按它们顶点的顺序形成或长或短的路(或圈),这就使得我们在实验中,可以对多条边同时处理。这样也就大大缩短了实验的时间。

在本文中,由于待求解的图的规模较大,所需要的编码较多,这就涉及了编码的质量和数量这一相互矛盾的指标。编码的质量越高,计算的可靠性越高,但是编码的数量反而越小。本文对已有编码的约束条件进行了改进,使得得到了较多的寡核苷酸序列,通过实验验证,这些序列能够保证DNA计算的可靠性。因此本文中提出的编码约束条件可用于其他相关的DNA计算中。

虽然,本章建立的DNA计算模型可用于处理较大规模的问题。然而因为随着问题规模的继续增大,所需要基本的DNA序列数目也会增大,从而导致DNA序列的长度不断增长。当达到一定的规模的时候,利用此DNA计算模型就可能会面临一些新的问题。在本项工作的研究过程中主要是实验过程中,遇到了假阳性现象。这也是DNA计算过程中对应的错误杂交的主要类型。也就是说如何用对定性的生化实验带来的误差给出可靠的定量分析是很重要的。这些问题都将作为我们进一步研究的目标。

致谢

感谢国家自然科学基金项目(61632002、61379059、61572046)的资助。

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.

Reference

- [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 $O(1.415^n)$ 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 $O(1.3289^n)$. *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.