粘贴 DNA 计算机模型(Ⅱ): 应用

许 进① 李三平② 董亚非① 魏小鹏③

(①华中科技大学分子生物计算机研究所, 武汉 430074; ②陕西师范大学数学与信息学科学院, 西安 710062; ③大连大学先进设计制造中心, 大连 116622. E-mail: jxu@mail.hust.edu.cn)

摘要 经典的粘贴 DNA 计算模型采用单、双链混合型 DNA 分子编码, 其生物操作具有无需 DNA 链的延伸、无需生物酶以及 DNA 链可重复使用等优点, 已经受到不同学科学者的关注. 在经典模型的基础上, 进行一定的扩展与完善, 必对 DNA 计算机的研究有良好的贡献. 基于此, 对粘贴 DNA 计算机模型进行了较为深入的研究: (1) 提出了基于粘贴模型的矩阵表达模型; (2) 对经典粘贴模型应用于图与组合优化等方面的研究成果给予综述, 诸如集合覆盖问题、图的顶点覆盖问题、图的 Hamilton 路与圈问题、图的团与独立集问题、图的生成树与 Steiner 树问题等; (3) 给出了基于粘贴模型的图的同构问题的算法.

关键词 **DNA** 计算 粘贴模型 ***k*-进制粘贴模型** 组合优化问题

在文献[1]中, 我们对粘贴DNA计算模型的基本思想、方法和理论进行了较为系统的论述. 在这篇文章中, 我们将主要应用文献[1]中给出的几种粘贴模型来建立求解图与组合优化中的一些NP-完全问题的DNA计算模型. 在已发表的成果中, 主要建立了如下几个NP-完全问题的DNA计算模型:

● 集合覆盖问题[2], 这是Roweis等人首次建立粘贴DNA计算模型的文章中, 应用文中的粘贴模型给出了集合覆盖问题的DNA计算模型;

● 图的顶点覆盖问题[3], 这个问题的粘贴DNA计算模型类似于文献[2]中的建立方法, 我们在本文中给予非常简要的介绍;

● 具有 20 个变量的可满足性(SAT)问题的DNA计算模型[4], 这是以Braich等人为首的Adleman研究组的成果, 他们在文献[4]中以Roweis等人建立的粘贴模型为工具来研究SAT问题. 这个实例的惟一解通过超过 1 百万(2^{20})次可能性的完全搜索之后被找 到. 这个问题的计算规模是迄今为止用非电子工具所解决的规模最大的一个问题. 这个规模似乎远远超过了独立于人类计算能力的正常范围;

● 文献[5]中给出了应用粘贴DNA计算的多个图论中的计算模型, 首先给出了图的边导出子图的粘贴计算模型, 然后在此模型的基础上, 建立了诸如图的Hamilton路与圈问题、Steiner数问题、图的团与图的独立集问题等的粘贴计算模型.

除上述已经获得的应用粘贴模型建立的一些NP-完全问题的 DNA 计算模型外, 在本文中还给出了我们应用全信息粘贴模型求解图的最大团与最大独立集、图的同构问题的 DNA 计算模型. 为了对这些问题给予较为清晰的论述, 我们在本文中给出了基于粘贴模型的 0-1 矩阵的表达, 相信这种矩阵表达对研制未来通用的 DNA 计算机是有用的.

为了本文的方便, 我们在此将粘贴模型中的四种基本操作: 合并、分离、设置与清除重述如下:

● 合并(merge): 将来自于两个试管T_1, T_2中的存储合成物组合在一个试管里, 所得到的试管可看作是由输入试管的并构成的多重集;

● 分离(separation): 对试管中的存储链, 由某位元上值的状态将其分离到两个试管中, 其中一个试管中的存储链在该位上的值为“1”, 而另一个试管中存储链在该位置上的值为“0”, 通过设计探针来实现. 更详细地讲, 即对于试管 T和整数 i, $1\leq i\leq n$, 产生两个新试管, 记为试管+(T, i)和试管–(T, i), 其中试管+(T, i)是由所有的第 i 个子链在试管 T 中均为双链的存储链构成, 而试管–(T, i)是由所有的第 i 个子链在试管 T 中均为单链的存储链构成;

● 设置(set): 将试管中所有的存储链某位元的值全部变成“1”, 即通过杂交将原来的序列段由单链变为双链. 更详细地讲, 对于试管 T 和整数 i, $1\leq i\leq n$, 操作设置产生一个新的试管 set(T, i), 使得试管 T 中每个存储合成物的第 i 个子链转入“开”. 对此, 如果它的第 i 个子链是“关”的, 则需一个适当的粘贴链被退火于一个存储链; 如果第 i 个子链总是“开”的, 则留下来的存储合成物不变;

● 清除(clear): 将试管中所有的存储链中某位的值全部变为“0”, 即通过加热将该序列由双链变为单链. 更详细地讲, 对于试管 T 和整数 i, $1\le i\le n$, 操作清除产生一个新的试管 clear(T, i), 使得该试管 T 中的每个存储合成物的第 i 个子链转入“关”, 对此, 最终的退火粘贴链被删除.

有关它们所对应的生物操作在此略去, 可参见文献[1, 2]. 且有关其他方面的未在本文中给出的概念、记号与定义, 如存储合成物等也可参见文献[1].

本文所言之图, 皆指无环、无圈的有限简单图, 文中一般地用$G = (V, E)$表示一个图, 其中V, E分别表示图G的顶点集和边集. 一个图G的子图H是指它的顶点集和边集分别满足$V(H)\subseteq V(G)$, $E(H)\subseteq E(G)$的图. 如果$V(H) = V(G)$, 则称H为G的一个生成子图. 设$V'\subseteq V(G)$, 则由V'导出的子图, 记作$G[V']$, 是图G的一个子图, 其顶点集边集分别为$V(G[V']) = V'$和$E(G[V']) = \{uv; u, v\in V', uv\in E(G)\}$. 所谓图$G$的(顶点)导出子图是指$V(G)$的某一个子集$V'$导出的子图. 与导出子图类似亦可定义边导出子图: 设E'是图G的一个非空边子集, 以E'为边集, 以E'中边的端点的全体为顶点子集所构造的图G的子图称为由E'导出的G的子图, 记作$G[E']$. 简称为边导出子图. 我们在本文中将会用到诸如图的团、独立集、Hamilton路与圈、导出子图等概念, 这些概念将会在适当的地方出现, 文中所用到未定义的有关图论中的概念与术语参见文献[6].

1 基于粘贴模型的矩阵表达

对于一个给定的 0-1 矩阵:

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

用存储合成物表示的方法是: 从左到右, 依次位置为$a_{11}, \cdots, a_{1n}, a_{21}, \cdots, a_{2n}, \cdots, a_{n1}, \cdots, a_{nn}$. 特别要强调的是: 这个次序是固定的. 若$a_{ij} = 1$, 则在$(i, j)$的位置上用双链来表示; 若$a_{ij} = 0$, 则在$(i, j)$的位置上用单链来表示. 如对于一个 3 阶矩阵:

$$A=\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

它所对应的存储合成物为如图 1 所示:

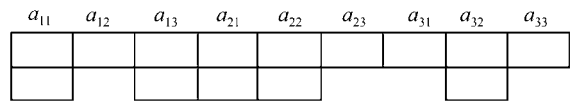

图 1 表示矩阵 A 的存储合成物

若A是对称的, 则从左到右的位置依次为$a_{11}, \cdots, a_{1n}, a_{22}, \cdots, a_{2n}, \cdots, a_{(n-1)(n-1)}, a_{(n-1)n}, a_{nn}$; 若$A$是对称的, 且对角线元素全为零时, 则从左到右的位置排列为$a_{12}, \cdots, a_{1n}, a_{23}, \cdots, a_{2n}, \cdots, a_{(n-1)n}$. 如对于如下的 4 阶矩阵

$$A'=\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

它的位置排序依次为$a_{12}, a_{13}, a_{14}, a_{23}, a_{24}, a_{34}$, 所对应的存储合成物如图 2 所示:

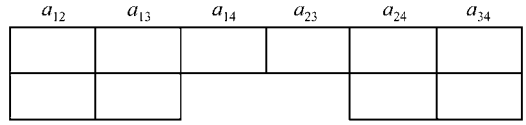

图 2 表示矩阵 A'的存储合成物

2 集合覆盖问题与图的顶点最小覆盖问题

本节主要介绍用粘贴模型求解图的集合覆盖问题的DNA计算模型[2]. 设$S = \{1, 2, \cdots, n\}$是具有n个元素的一个非空集合, $C = \{C_1, C_2, \cdots, C_m\}$是由$S$的$m$个非空子集构成的集族, 且满足: $\bigcup_{i=1}^{m} C_i = S$, 则称$C$是集合$S$的一个覆盖. 所谓集合$S$的最小覆盖问题, 是指从$C$中找出集合数目最小的集合族来覆盖$S$, 并把这个最小的数目称为最小覆盖数.

若图 G 中的一个顶点和一条边相互关联, 则称它们相互覆盖. 设 $S\subseteq V$, 如果 G 的任意一条边都至少有一个端点属于 S, 则称 S 为 G 的一个顶点覆盖; 若 S 是 G 的顶点覆盖, 且对任一顶点 $v\in S$, $S-\{v\}$不再是图 G 的一个顶点覆盖, 则称 S 为 G 的极小顶点覆盖; 若 S 是顶点覆盖, 且 G 不存在另一顶点覆盖 S'满足 $|S'|<|S|$, 则称 S 是图 G 的最小顶点覆盖.

众所周知, 求解集合最小覆盖问题、图的顶点覆盖问题都是困难的 NP-完全问题. 图的顶点覆盖问题(Vertices Covering Problem, VCP)是指找给定图中顶点集的一个最小子集, 使得它覆盖给定图中 G 的所

有边. 这两个问题在诸如分子生物序学、调度问题、信息检索、错误诊断和恢复、集装线平衡、油轮行程安排以及开关理论等有着广泛的应用[7~14].

关于求解集合覆盖问题与图的顶点覆盖问题的研究成果很多, 如文献[15]给出了一种启发式算法来求解图的最小顶点覆盖问题; 一些数学工作者已经证实对于特殊的图, 通过采用特殊的算法, 可以在多项式时间内求得图的子集. 基于人工神经网络的图的顶点覆盖问题的算法及研究进展参见文献[16].

Roweis等人在文献[2]中用粘贴模型给出了求解最小集合覆盖问题一种DNA计算模型. 关于此模型的基本思想是: 用具有单链、双链混合的存储合成物来表示集合C中全部 2^m种可能出现的选择. 当子集C_i被选中时, 则对i进行标记. 然后, 将那些标记着所含的全部n种物件的存储合成物分开, 并且读出使用最少子集合数的那一条. 其具体的算法要点设计如下:

(1) 关于存储链的设计: 存储链的位段共有$l = m+n$个, 其中前m个依次表示子集合C_1, C_2, …, C_m, 我们称这前m位为主链(参见文献[1]); 后n个依次表示集合元素 1, 2, …, n的位段, 我们称这后n位为辅助链, 或简称为辅链. 每个位段由长度为k的DNA碱基序列构成;

(2) 初始试管T_0的设计: 设计的数据库为$(m+n, m)$, 这个数据库中含有全部的 2^m个存储合成物, 即在主链中的每位粘贴链的存在与否的各种可能性在初始试管T_0中都存在, 有关初始试管T_0设计的生物技术参见文献[2];

(3) 对初始试管T_0中的每一条存储合成物的辅链进行标记: 标记后的结果是: 使得该条存储合成物的主链上每个双链位段对应的子集合C_i中所含集合S中的每个元素在标记成双链. 其具体的生物操作是:

① 对于每个i, $i = 1, 2, \cdots, m$, 施行分离操作: 将试管T_0分离成试管$+(T, i)$和试管$-(T, i)$;

② 对$+(T, i)$中每个$C_i = \{i_1, i_2, \cdots, i_t\}$中的每个元素$i_j, j = 1, 2, \cdots, t$, 施行设置操作: $\mathrm{set}(+(T_0, i), m+i_j)$;

注 对初始试管T_0施行上述生物操作后, 将所得到的新的试管$+(T, i)$与试管$-(T, i)$混合在一起, 所得到的试管仍记为T_0.

(4) 保留辅链全为双链的存储合成物, 清除其余的存储合成物: 方法是: 对辅链中的每个位段施行分离操作, 保留双链, 清除单链. 具体的生物操作如下:

① 对于每个$m+j, j = 1, 2, \cdots, n$, 施行分离操作: 将试管T_0分离成试管$+(T, m+j)$和试管$-(T, m+j)$;

② $T_0 \leftarrow +(T_0, m+j)$, 清除$-(T_0, m+j), j = 1, 2, \cdots, n$.

(5) 寻找最小覆盖集: 方法是: 对T_0中的存储合成物按其所含的子集合数分别标记为$T_i (i \geq 1)$, 按i的大小顺序来检测T_i, 第 1 次检测到的非空试管极为最小集合覆盖的子集合的个数.

(6) 检测具体的最小覆盖集: 方法是: 对T_i中的存储合成物进行检测, 识别主链中的每个位段的单、双性. 这个过程可通过荧光标记的探针来完成.

上述算法取 $O(mn)$步, 输入 $O(mn)$个位点.

文献[3]中利用Roweis等人在文献[2]中所提出的粘贴模型给出了另一种覆盖问题, 图的顶点覆盖问题的粘贴DNA计算模型. 在该文章中, 从算法设计、算法的形式化程序、算法的生物实现等通过一个具体的例子给予较为详细地讨论, 限于篇幅, 我们在此略去, 有兴趣的读者可参见原文.

3 可满足性(SAT)问题

可满足性(SAT)问题是一个著名的 NP-完全问题, 它在逻辑电路的设计及密码问题的研究中都有广泛的应用. 通常 SAT 问题可以表述为: 给定一个 Boole 表达式:

$$F = C_1 \wedge C_2 \wedge \cdots C_m,$$

其中$C_i = v_1 \vee v_2 \vee \cdots v_n$, $v_i (i \in \{1, 2, \cdots, m\})$为Boole变量取值为 0 和 1, "$\wedge$"为逻辑与, "$\vee$"为逻辑或. SAT问题就是求满足$F = 1$ 的所有Boole变量v_i的真值分配表. 显然, 对于包含n个变量的Boole表达式共有 2^n个不同的真值赋值.

由于SAT问题的重要性与可能性, 因而关于这方面的研究成果很多, 我们在此只就基于DNA计算的SAT问题的一些主要模型给予介绍. 1995 年, Lipton对于具有 3-可满足性问题(SAT问题)给出了一种DNA计算模型[18], Landweber等于 1999 年提出了一种基于RNA的"破坏性"的SAT问题的算法 [19]. 2000 年Sakamoto等巧妙的运用单链DNA分子的"发夹"结构给出了一个 3-SAT问题的算法[20]; 2000 年, Liu等给出了一种基于表面的SAT问题的算法[21]; 同年, Dirk等人用RNA分子代替DNA分子给出了SAT问题的一种实验性的计算模型, 并讨论国际象棋问题的RNA计算模型[22]. 2003 年, 我们提出了基于分子信标技术的SAT问题的一种DNA计算模型[23].

2002 年, Braich等人应用粘贴模型等给出了具有 20 个变量的SAT问题的DNA计算模型, 并通过生化实验, 获得了成功, 这是迄今为止运算量最大的一个DNA计算实例[4].

文献[4]中所给出的含有 20 个变量的 3-可满足性问题Φ 如下:

$$\Phi=(\bar{x}_3\vee\bar{x}_{16}\vee x_{18})\wedge(x_5\vee x_{12}\vee\bar{x}_9)\wedge(\bar{x}_{13}\vee\bar{x}_2\vee x_{20})\wedge$$
$$(\bar{x}_9\vee\bar{x}_5\vee x_{12})\wedge(\bar{x}_4\vee x_6\vee x_{19})\wedge(x_5\vee x_{17}\vee x_9)\wedge$$
$$(\bar{x}_1\vee x_4\vee\bar{x}_{11})\wedge(\bar{x}_{19}\vee\bar{x}_2\vee x_{13})\wedge(x_5\vee x_{17}\vee x_9)\wedge$$
$$(x_{15}\vee x_9\vee\bar{x}_{17})\wedge(\bar{x}_5\vee\bar{x}_9\vee\bar{x}_{12})\wedge(x_6\vee x_{11}\vee x_4)\wedge$$
$$(\bar{x}_{15}\vee\bar{x}_{17}\vee x_7)\wedge(\bar{x}_6\vee x_{19}\vee x_{13})\wedge(x_{12}\vee\bar{x}_9\vee x_5)\wedge$$
$$(x_{12}\vee x_1\vee x_{14})\wedge(x_3\vee x_{20}\vee x_2)\wedge(x_{10}\vee\bar{x}_7\vee\bar{x}_8)\wedge$$
$$(\bar{x}_5\vee x_9\vee\bar{x}_{12})\wedge(x_{18}\vee\bar{x}_{20}\vee x_3)\wedge(\bar{x}_{10}\vee\bar{x}_{16}\bar{\vee}x_{18})\wedge$$
$$(x_1\vee\bar{x}_{11}\vee\bar{x}_{14})\wedge(x_8\vee\bar{x}_7\vee\bar{x}_{15})\wedge(\bar{x}_8\vee x_{16}\vee\bar{x}_{10}),$$

该文中有意设计出Φ具有惟一满足真值分配, 这个惟一满足真值分配的解如下:

$$x_1=\mathrm{F},\ x_2=\mathrm{T},\ x_3=\mathrm{F},\ x_4=\mathrm{F},\ x_5=\mathrm{F},\ x_6=\mathrm{F},$$
$$x_7=\mathrm{T},\ x_8=\mathrm{T},\ x_9=\mathrm{F},\ x_{10}=\mathrm{T},\ x_{11}=\mathrm{T},\ x_{12}=\mathrm{T},$$
$$x_{13}=\mathrm{F},\ x_{14}=\mathrm{F},\ x_{15}=\mathrm{T},\ x_{16}=\mathrm{T},\ x_{17}=\mathrm{T},$$
$$x_{18}=\mathrm{F},\ x_{19}=\mathrm{F},\ x_{20}=\mathrm{F}.$$

文献[4]中所使用的模型是相关于由Roweis等所提出的粘贴模型[2]. 粘贴模型在计算中使用了两种基本的操作: 在子序列上的分离, 以及粘贴的应用. 只有分离操作应用于最近的研究中. 分离通过使用固定在充满凝胶聚丙烯酰胺的玻璃存储体的核苷酸探针来实现. 携带信息DNA串使用电泳在这些存储体里移动. 使其与固定探针发生杂交反应, 且被保留在存储体中; 然后, 使其在高温下融解, 通过电泳来从这些探针中释放俘获的串. 被释放的串为了进一步的分离, 可以通过电泳传送到新的模块中去.

利用电泳在充满凝胶的玻璃模块之间传送 DNA 串, 导致一种"干的"且可自动化的计算机. 由于在分离过程中, 共价键要么形成, 要么断裂, 所以 DNA 串和玻璃模块对于多次计算中可以再度使用.

Braich等在文献[4]中所用的(数据)库采用了Lipton在文献[18]中编码方法. 对这 20 个变量中的每个变量$x_k(k=1,\cdots,20)$, 设计了两个不同的长度为 15 的碱基"值序列": 其中一个表示真的(T), 记做 X_k^{T}; 另一个表示假的(F), 记做 X_k^{F}; 其具体编码如下:

X_1^{T} = TTA CAC CAA TCTCTT,
X_1^{F} = CTC CTA CAA TTC CTA;
X_2^{T} = ATT TCCAAC ATA CTC,
X_2^{F} = AAA CCT AAT ACT CCT;
X_3^{T} = TCA TCC TCT AAC ATA,
X_3^{F} = CCC TAT TAA TCAATC,
X_4^{T} = TCA CTC CAC TTA ACT,
X_4^{F} = TAC TTATAA CTT CCC,
X_5^{T} = ATA ACC ACA AAC TCA,
X_5^{F} = TCT CAA TAC CAC CTA,
X_6^{T} = CTA TCC AATAAC CTC,
X_6^{F} = TTC ATA CAC TTA CAC,
X_7^{T} = TTCCAC CCC AAT AAA,
X_7^{F} = AAC TCA TAC TAC TCA,
X_8^{T} = CTA TTT ATA TCC ACC,
X_8^{F} = TAT TCT CACCCA TAA,
X_9^{T} = ACA CCT AAC TAA ACT,
X_9^{F} = ACA CTA TCA ACA TCA,
X_{10}^{T} = CTA CCC TAT TCTACT,
X_{10}^{F} = CCT TTA CCT CAA TAA,
X_{11}^{T} = ATCTTT AAA TAC CCC,
X_{11}^{F} = CTC CCA AAT AAC ATT,
X_{12}^{T} = TCC ATT TCT CCA TAT,
X_{12}^{F} = AAC TTCACC CCT ATA,
X_{13}^{T} = TTT CTT CCA TCA CAT,
X_{13}^{F} = TCA TAT CAA CTC CAC,
X_{14}^{T} = CAT TCAATC CAC TAC,
X_{14}^{F} = ACC CAA TCC TCT TAA,
X_{15}^{T} = AAC AAC CTT ATC CTT,
X_{15}^{F} = TAA TAACCC ATC CTA,
X_{16}^{T} = TCA CTA CAT TAC CTT,
X_{16}^{F} = TCA TCA AAC CTC ACA,
X_{17}^{T} = ACA AACCCT AAC ATT,
X_{17}^{F} = CTC AAC AAT TTT CCA,
X_{18}^{T} = TCT TAC CAT CTT CAT,
X_{18}^{F} = AAC ACATTA CTT CCT,
X_{19}^{T} = CTC TTC TCC TCT TTT,
X_{19}^{F} = ACC CAT TAC TAC CAT,
X_{20}^{T} = ACA CAAATA CAC ATC,
X_{20}^{F} = CAA CCA AAC ATA AAC.

对$k=1,\cdots,20$, $Z=$ T或F, 用 $\overline{X_k^Z}$ 来表示 X_k^Z 的Watson-Crick互补链. 2^{20}个真值分配中的每一个真值分配用长度为 300 的碱基"库序列"来表示, 其中每个长度为 300 个碱基的构成是: 对每个变量取一个值序列(即取一个长度为 15 的碱基序列), 然后依次连接起来即可. 在库中的单链DNA分子称为"库链". 所有配成互补链的库链的集合称为"完整库".

Braich 等人所设计的计算机由具有一个热室和一个冷室的电泳箱; 一块充满聚丙烯酰胺凝胶体的玻璃"库模板", 其中在凝胶体上含有共价键探针全库、且对Φ中 24 个子句中的每个子句, 有一块充满

聚丙烯酰胺凝胶的玻璃“子句模块”，其中在凝胶体上含有共价键探针，且被设计的只接收满足该子句的真值分配编码所对应的库链，如图 3 所示:

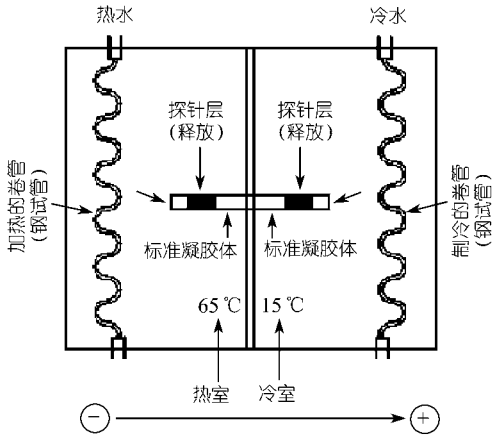

图 3　Braich 等所设计的基于粘贴计算模型的 DNA 计算机示意图

这台基于粘贴模型的 DNA 计算机的具体计算步骤如下:

步骤 **1**　将库模块插入电泳箱中的热室中，同时把第 1 个子句模块插入冷室. 然后进行电泳;

步骤 **2**　从电泳箱中移去这两个模块. 弃掉热室中的模块，清洗电泳箱并加入新的缓冲液. 从冷室里把模块取出来加到热室，且将下一个子句的模块加入到冷室，开始电泳;

步骤 **3**　对剩余的 22 个子句中的每一个重复施行步骤 2;

步骤 **4**　从最后一个子句的模块中取出结果链，进行 PCR 扩增，“读出”答案.

限于篇幅，有关最终的惟一解的检测等的详细地的生物处理技术见文献[4]. 这里略去.

4　Zimmermann的研究成果[5]

Zimmermann在文献[5]中对计算所有k-团，独立k-集，Hamilton路与圈，关于给定边集或者顶点集的steiner树提出了基于粘贴模型的DNA计算模型. 这些模型不仅确定了解的存在而且产生了所有的解. 对于具有n个顶点m条边的无向图G，这些算法模型的运行时间在$n+m$之内，是线性的. Zimmermann在他的研究中对这些粘贴算法使用了小组合输入库代替了通常所用的大库. 该文中所描述的算法事实上完全是理论上的，没有生物实验. 在齐默曼的小组合输入库中，主要是利用了“图的边导出子图的粘贴计算形式化算法程序”和“赋权(Weightening)算法程序”，其中赋权算法程序是针对初始试管N_0中，每个存储合成物的辅链中恰好有k个是双链的算法程序. 下面，我们从这两个基本程序出发来展开对Zimmermann工作的介绍.

4.1　两个基本算法程序

k-边导出子图: 设$G = (V, E)$是一个有限无向图，其顶点集和边集分别为$V = \{v_1, v_2, \cdots, v_n\}$与$E = \{e_1, e_2, \cdots, e_m\}$. 其中第$i$条边用$e_i = (v_{i1}, v_{i2})$来表示. 下面的粘贴算法给出了产生图$G$的所有$k$-条边的导出子图的算法形式化的程序:

1. 边导出子图(N_0, m, n)
2. for $i\leftarrow 1$ to m do
3. 分离$+(N_0, i)$与$-(N_0, i)$
6. $N^+\leftarrow$set$(+(N_0, i), m+i_1)$
7. $N^+\leftarrow$set$(N^+, m+i_2)$
8. $N_0\leftarrow$合并$(N^+, -(N_0, i))$
9. od
9. 返回N_0
10. 结束边导出子图.

由前述易看出，该算法并行地考虑存储合成物中辅链的第i位段，$1\le i\le m$. 对于哪些在第i个位上是“开”的位段，则图G的边e_i出现在相应的k-边子集中. 如果第i个位段是“开”的，则表示边e_i的子位段$m+e_1$和$m+e_2$等也处于“开”的状态. 在最终的试管里，存储复合物对应的是由所给的k-边子集导出的图G的子图. 该算法需要 $4m$步.

赋权(Weightening)算法程序: 本算法的功能是: 在输入初始试管N_0中选择n个位段的辅链中恰好有k个是“开”的存储合成物的算法. 具体的形式化的算法如下:

1. Weightening (N_0, m, n, k)
2. for $i\leftarrow 0$ to $n-1$ do
3. for $j\leftarrow i$ down to 0 do
4. 分离$+(N_j, m+i+1)$和$-(N_j, m+i+1)$
5. $N_{j+1}\leftarrow$merge$(+(N_j, i+1), N_{j+1})$
6. $N_j \leftarrow -(N_j, i+1)$
7. od
8. od
9. 返回N_k
10. 结束 Weightening

由上述图的边导出子图粘贴算法，很容易推出基于粘贴计算的图的生成树的计算模型，这个模型

实际上是将存储合成物与图的生成树之间建立起 1-1 对应的关系. 具体如下:

1. 生成树 (N_0, m, n)
2. N_0 ←—— 边导出子图 (N_0, m, n)
3. for i ←—— 1 to n do
4. 分离 $+(N_0, m+i)$ 与 $-(N_0, m+i)$
5. N_0 ←—— $+(N_0, m+i)$
6. od
7. 返回 N_0
8. 结束生成树

该算法需要 $4m+n$ 步.

4.2 *k*-团与 *k*-独立集

图的团与独立集的概念实际上是完全等价的: 设V'是图G的一个顶点子集, 如果由V'导出的顶点子图$G[V']$是图G的一个完全子图, 则称V'是图G的一个团, 若$|V'| = k$, 即V'中含有k个顶点, 则称V'是图G的一个k-团; 设V'是图G的一个顶点子集, 如果由V'导出的顶点子图$G[V']$是图G的一个完全空图, 则称V'是图G的一个独立集, 若$|V'| = k$, 则称V'是图G的一个k-独立集. 显然, V'是图G的一个k-独立集当且仅当V'是图G的补图$\overline{G}$一个k-团. 因而, 求解一个图的一个k-团问题完全等价于求解一个图的一个k-独立集团问题. 图的最大独立集问题不但在工程技术上有直接或者间接的应用, 而且在数学理论本身具有良好的应用[16]. 因此, 给出具有快速准确的图的最大团或者最大独立集算法意义重大. 关于这方面的研究结果很多, 可由文献[16]中获知. 基于等价性, 具体地讲, 我们在此给出求解一个图的一个k-团问题的算法即可.

1. k-团(N_0, m, n, k)
2. N_0←边导子图(N_0, m, n)
3. N_0←Weightening(N_0, m, n, k)
11. 如果N_0非空则
12. 返回N_0
13. 否则
14. 输出"无 k-团"
15. 结束 k-团计算

该算法所需要的算法步骤为 $4m+2n(k+1)-k^2-k$.

4.3 Hamilton 路与圈问题的粘贴 DNA 计算模型

众所周知, 图中的路与圈问题是图论学科中最核心的问题之一. 在图论中长时间未得到解决的问题之一是寻找一个实用且简单的Hamilton图的特征. 目前人们只能对那些比较特殊的图, 或者在某些参数限制条件下给出判别图的Hamilton性. 判别一个图是否是Hamilton图是一个困难的NP-完全问题, 且具有广泛的应用背景, 因而受到许多不同领域内学者的关注. 关于这方面的研究进展可参见文献[24]中的第五章. 在这一小节里, 我们将介绍Zimmermann[5]给出的基于粘贴模型的Hamilton路与圈的形式化程序算法.

图的 Hamilton 路问题粘贴计算模型的形式化算法:

1. Hamilton路(N_0, m, n)
2. N_0←边导子图(N_0, m, n)
3. N_0←生成树(N_0, m, n)
4 for i← 1 to m do
5. 分离$+(N_0, i)$和$-(N_0, i)$
6 for j←1 to 2 do
7 分离$+(+(N_0, i), m+n+i_j)$和$-(+(N_0, i), m+n+i_j)$
8. N^+←set $-(+(N_0, i), m+n+i_j)$
9. N^-←clear $+(+(N_0, i), m+n+i_j)$
10. od
11. N_0←合并$(-(N_0, i), N^+, N^-)$
12. od
13. N_0←Weightening$(N_0, m+n, n, 2)$
14. 如果非空, 则
15. 返回N_0
16. 否则
17. 输出"无 Hamilton 路"
18. 结束 Hamilton 计算

该算法需要 $4m+n+8m+6n-6 = 12m+7n-6$ 步.

图的 Hamilton 圈问题粘贴模型的形式化算法:

1. Hamilton圈(N_0, m, n)
2. N_0←Hamilton路(N_0, m, n)
3. for i←1 to m do
4. 分离$+(N_0, m+n+i_1)$和$-(N_0, m+n+i_1)$
5. 分离$+(+(N_0, m+n+i_1), m+n+i_2)$和 $-(+(N_0, m+n+i_1), m+n+i_2)$
6. N^+←merge$(N^+, +(+(N_0, m+n+i_1), m+n+i_2))$
7. N_0←merge$(-(N_0, m+n+i_1), -(+(N_0, m+n+i_1), m+n+i_2))$
8. od
9. 如果N^+非空, 则
10. 返回N^+
11. 否则

12. 输出“无 Hamilton 圈”
13. 结束 Hamilton 圈的计算

该算法需要 $12m+7n-6+4m = 16m+7n-6$ 步.

4.4 Steiner 树问题的粘贴计算模型

给定平面上的n个顶点$v_1, v_2, \cdots, v_n$, 要求寻找一个包含这n个顶点的最短网络N. 这就是所谓的网络型的Steiner问题. 也是Steiner树问题提出的最初模型. 现已提出了多种类型的Steiner树问题, 主要有诸如Euclid型的Steiner树问题、图的Steiner树问题、直线型Steiner树问题以及定向制Steiner树问题. 它们不但都是困难的NP-完全问题, 且具有良好的应用背景, 因而备受学者们的关注. 关于Steiner树的有关基本理论与研究进展可参见文献[25]. 在这一节里, 我们只对图的Steiner树问题给出粘贴计算模型.

设 Z 是给定的图 G 的一个顶点子集, $H = (U, F)$ 是图 G 的一棵子树. 如果 $Z \subseteq U$, 且 H 是图 G 中包含 Z 的边数最少的一个子图, 由于 H 是一棵树, 故称为它是图 G 的 Steiner 树. 下面给出了形式化的算法:

1. Steiner树(N_0, m, n, l)
2. N_0←边导子图(N_0, m, n)
3. for i←1 to n do
4. 分离+(N_0, $m+i$)和–(N_0, $m+i$)
5. N_0←+(N_0, $m+i$)
6. od
7. for i←1 to l do
8. 分离+(N_0, $m+i$)和–(N_0, $m+i$)
9. N_0←+(N_0, $m+i$)
10. od
11. 如果N_0非空则
12. 返回N_0
13. 否则
14. 输出“无 Steiner 树”
15. 结束 Steiner 树计算

此算法需要 $4m+2n+2l$ 步.

该算法给出了确定在图 G 中包含 Z 的所有的 k-边导出子图的 Steiner 树的算法.

在文献[5]中, 作者还给出了基于粘贴算法的k-个顶点子集导出的图G的子图的求解算法, 由于这个问题是一个意义不太大的问题, 且在应用于其他困难NP-完全问题上没有太大的应用, 故在此未给予介绍, 有兴趣的读者可参见原文.

5 图的同构问题的粘贴计算模型

两个图G与G', 称为是同构的, 如果存在一个从$V(G)$到$V(G')$之间的保持相邻性的 1-1 映射, 换言之, 存在 1-1 映射: σ: $V(G)\to V(G')$, 且对$\forall u$, $v\in V(G)$, $uv\in E(G)$当且仅当$\sigma(u)\sigma(v)\in E(G')$, 我们称$\sigma$是从$G$到$G'$的一个同构映射. 全体从$G$到$G'$的同构映射构成的集合记作$I(G, G')$.

图的同构问题一直受到数学界与工程技术界, 特别是大系统建模技术人员的关注, 其原因主要来自两个方向: 第一, 从理论上讲, 图的同构问题是困难的算法问题, 目前还不知道它是P-问题还是NP-完全问题; 第二, 图的同构问题具有很好的应用背景, 特别是应用于系统建模: 如果建模者能够证明需建的模型与已有的某模型同构, 则勿需再建, 这将大大节省人力物力. 传统的方法检验图的同构问题是非常困难的, 特别是当图的顶点数较大时, 几乎是不可能的. 正因为如此, 目前关于图的同构问题的研究成果很多[16,26]. 但是, 至今尚未见到建立基于DNA计算模型的图的同构算法模型. 下面, 我们将以图 4 中的两个图为例给出用于图的同构计算的粘贴计算模型. 一般地我们令G, G'是具有相同顶点数与边数的两个n阶图. 其中$V(G) = \{x_1, x_2, \cdots, x_n\}$, $V(G') = \{y_1, y_2, \cdots, y_n\}$.

下面, 我们主要对基于粘贴计算的图的同构问题 DNA 计算模型的主链、辅链和决策链的设计要点给予介绍, 其他详细内容参见另文[1)].

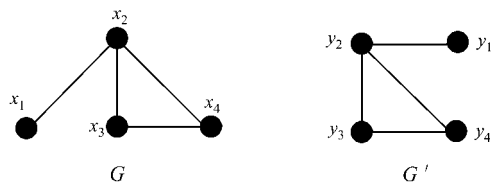

图 4　两个同构的 4-阶图

总位段的设计: 主链、辅链和决策链, 其中主链由全体同构映射集合构成; 辅链由两个图的相邻矩阵构成; 决策链用于判断同构映射是否保持相邻性;

(1) 主链的设计: 主链由全体同构映射构成. 主链的位段数总共$\frac{1}{2}n(n+1)$: $x_i(y_j)$, $i\neq j$, $i\leq j$, $i, j = 1,$

1) Xu Jin. A Full-messages sticker DAN computing model

2, …, n. 对于图 4 中所示的两个图 G, G', 其主链位段是:

$$x_1(y_1), x_1(y_2), x_1(y_3), x_1(y_4); x_2(y_2),$$
$$x_2(y_3), x_2(y_4), x_3(y_3), x_3(y_4), x_4(y_4).$$

如图 5 所示, 这里要说明的是主链中用ij来代表$x_i(y_j)$, 其中$i, j = 1, 2, 3, 4$.

(2) 辅链的设计: 设计了两个辅链, 分别表示两个图G, G'相邻矩阵, 并分别称为辅链G、辅链G', 如图 5 所示, 其中辅链G中的ij表示了图G中的两个顶点v_i与v_j之间的相邻关系: 若相邻, 则为双链, 否则为单链(如图 5 所示); 辅链G'中的ij表示了图G'中的两个顶点u_i与u_j之间的相邻关系, 其表示方法与前面的类似. 其中图G的相邻矩阵为

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

由于图的相邻矩阵 $A(G)$是一个对角线所有的元素均为零, 且对称的矩阵, 所以, 只要设计 $\frac{1}{2}n(n-1)$ 个位段即可;

(3) 决策链的设计: σ图G到图G'的同构映射当且仅当对于图G的每一对顶点v_i, v_j, v_i与v_j在图G中相邻有且只有 $\sigma(v_i)$ 与 $\sigma(v_j)$ 在图 G' 中相邻. 设 v_i, $v_j \in V(G)$, 令$\sigma(v_i) = v_i'$, $\sigma(v_j) = v_j'$, 则$v_iv_j \in E(G)$当且仅当$v_i'v_j' \in E(G')$. 基于此, 我们在决策链的设计中, 用ij表示$v_iv_j \in E(G)$与$v_i'v_j' \in E(G')$的真伪关系: 若这两个关系均成立, 或者均不成立(即 $v_iv_j \in E(G)$ 且 $v_i'v_j' \in E(G')$, 或者 $v_iv_j \notin E(G)$ 且 $v_iv_j \notin E(G)$), 则ij对应双链, 若其中一个成立, 另外一个不成立(即 $v_iv_j \in E(G)$ 但 $v_i'v_j' \notin E(G')$; 或者 $v_iv_j \notin E(G)$ 但 $v_i'v_j' \in E(G')$), 则 ij 对应单链. 这种关系可用如下表 1 给予说明, 其中"1"表示属于关系, 即$v_iv_j \in E(G)$, 或者$v_i'v_j' \in E(G')$, "0"表示不属于关系, 即$v_iv_j \notin E(G)$或者 $v_i'v_j' \notin E(G')$.

表 1 说明决策状态表

$v_iv_j \in E(G)$	$v_i'v_j' \in E(G')$	决策链的状态
0	0	双链
0	1	单链
1	0	单链
1	1	双链

例如, 如果我们选取

$$\sigma = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_2 & y_1 & y_3 & y_4 \end{pmatrix}$$

即在主链上设置 12, 21, 33, 44 为双链, 即对初始试管T_0实施生物操作: set(T_0, $x_1(y_2)$), set(T_0, $x_2(y_1)$), set(T_0, $x_3(y_3)$), set(T_0, $x_4(y_4)$), 如图 6 所示.

在此基础上, 我们来设置决策链: 由于$x_1x_2 \in E(G)$且$\sigma(x_1)\sigma(x_2) = y_2y_1 \in E(G')$, 故在决策链中位段 12 为双链; 其他关系在表 2 中给出:

由此表可给出决策链的设置: set(T_0, 12), set(T_0, 34), 于是, 由图 6 中所给出的试管T_0, 我们可以在此两个生物操作的基础上, 得到新的试管T_0如图 7 所示.

表 2 映射σ 所对应的决策区

i, j 的取值	$x_ix_j \in E(G)$	$\sigma(x_i)\sigma(x_j) \in E(G')$	决策位段	决策状态
$i = 1, j = 2$	1	1	12	双链
$i = 1, j = 3$	0	1	13	单链
$i = 1, j = 4$	0	1	14	单链
$i = 2, j = 3$	1	0	23	单链
$i = 2, j = 4$	1	0	24	单链
$i = 3, j = 4$	1	1	34	双链

由于决策链中存在单链, 故这个映射σ不是同构映射, 即$\sigma \notin I(G, G')$.

6 结论

本文是在理论部分的基础上, 对粘贴模型的进一步扩展与应用. 主要内容如下:

(1) 给出了基于粘贴模型的 0-1 矩阵的表示, 这将有助于进一步扩大粘贴 DNA 计算机模型的应用范围与实用性;

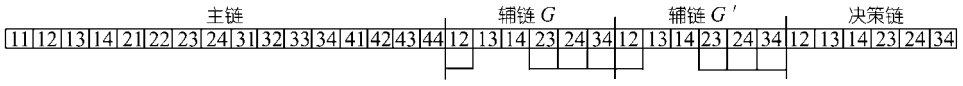

图 5 图 4 中所示图 G, G'的同构存储链

图 6 主链的设置

图 7 决策链的设置

(2) 讨论了求解图与组合优化中的一些 NP-完全问题的 DNA 计算模型这些模型涉及诸如集合覆盖问题、图的顶点覆盖问题、图的团与最大团以及图的独立集问题、可满足性问题、Steiner 树问题、Hamilton 路与圈问题等;

(3) 给出了基于粘贴模型的图的同构算法模型, 特别重点给出了有关主链、辅链和决策链的设计问题等.

粘贴模型的研究应属于初始阶段, 有许多要解决的问题. 从目前的研究进展来看, 用电子计算机参与控制是一个好的思想, 另外, 将诸如连接酶、核酸内切限制酶等与分子信标等荧光技术程序化的加入粘贴 DNA 计算模型, 使此模型进一步的通用化. 我们以为, 这是一个很有前途的 DNA 计算机研究方向.

致谢 审稿专家提出了几个很有建设性的意见, 指出了本文的原稿中几处错误, 并提出问题缺陷的解决方法, 作者在此表示深深的感谢. 本工作为国家自然科学基金(批准号: 60174047, 60103021, 60274026)、博士点基金及湖北省自然科学基金资助项目.

参 考 文 献

1 许进, 董亚非, 魏小鹏. 粘贴DNA计算机模型(Ⅰ): 理论. 科学通报, 2004, 49(3): 205~212[摘要] [PDF]
2 Roweis S, Winfree E, Burgoyne R, et al. A sticker based arcchtecture for DNA computation. In: Baum E B, eds. DNA Based Computers, Proc 2nd Annual Meeting, Princeton, 1999. 1~27
3 Gao Lin, Xu Jin. DNA solution of vertex cover problem based on sticker model. Chinese Journal of Electronics, 2002, 11(2): 280~284
4 Braich Ravinderjit S, Nickolas Chelyapov, Cliff Johnson, et al. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296: 499~502[DOI]
5 Zimmermann Karl-Heinz. Efficient DNA sticker algorithms for NP-complete graph problems. Computer Physics Communications, 2002, 144: 297~309[DOI]
6 Bondy J A, Murty U S R. Graph Theory with Applications. London, Basingtoke, New York: The Macmillan Press LTD, 1976
7 Papadimitriou C H, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. Englewood Cliffs, N J : Prence Hall, 1982. 358~409
8 Tinhofer G. Computational Graph Theory. Vienna: Springer- Verlag, 1990
9 Golumbic M C. Algorithmic Graph Theory and Peffect Graphs. New York: Academic Press, 1980
10 Vinnakota B, Andrews J. Repair of RAMs with clustered faults. In: Proc Int'l Conf Computer-Aided-Design, 1992. 582~585
11 Paias A, Paixao J. State space relaxation for set covering problem related to bus driver scheduling. Eur J Opl Res, 1993, 71: 303~316[DOI]
12 Beasley J E, Jornsten K. Enhancing an algorithm for set covering problems. Eur J Opl Res, 1992, 58: 293~300[DOI]
13 Lorena L A N, Belo Lopes F. A surrogate heuristics for set covering problems. Eur J Opl Res, 1994, 79: 138~150[DOI]
14 Fisher M L, Kedia D. Optimal solution of set covering/partitioning problems using dual heuristics. Mgmt Sci, 36: 674~688
15 Naft J. Neuropt: Neurocomputing for multiobjective design optimization for printed circuit board component. In: Proc Joint Conf Neural Networks, 1989. 503~506
16 许进, 保铮. 神经网络与图论. 中国科学, E辑, 2001, 31(6): 533~555[摘要] [PDF]
17 Leonard M Adleman. Molecular computation of solutions to combinatorial problems. Science, 1994, 266(11): 1021~1023
18 Lipton Richard J. DNA solution of hard computational problems. Science, 1995, 268(28): 542~545
19 Cukras A R, Faulhammer D, Lipton R J, et al. Chess games: A model for RNA-based computation. Biosystems, 1999, 52: 35~45[DOI]
20 Sakamoto K, Gouzu H, Komiya K, et al. Molecular computation by DNA hairpin formation. Science, 2000, 288: 1223~1226[DOI]
21 Liu Q, Wang L, Frutos A G, et al. DNA computing on surfaces. Nature, 2000, 403: 175~179[DOI]
22 Faulhammer Dirk, Cukras Anthony R, Lipton Richard J, et al. Molecular computation: RNA solutions to chess problems. Biochemistry, 2000, 97(4): 1385~1389
23 殷志祥, 张凤月, 许进. 基于分子信标的 DNA 计算. 生物数学学报, 2003, 18(4): 1~5
24 许进. 自补图理论及其应用. 西安: 西安电子科技大学出版社, 1999
25 Hwang F K, Richards D S, Winter P. The Steiner Tree Problem. New York: Elsevier Science Publishers B V Press, 1992
26 许进, 张军英, 保铮. 基于 Hopfield 神经网络的图的同构算法. 电子科学学刊, 1996, 图论专集: 116~121

(2003-09-29 收稿, 2003-12-18 收修改稿)