

图 3-着色问题的 $O(2^n)$ 链数 DNA 计算机算法

李肯立¹, 周 旭¹, 许 进²

(1. 湖南大学计算机与通信学院, 湖南长沙 410082; 2. 华中科技大学分子生物计算机研究所, 湖北武汉 430074)

摘 要: 随着 DNA 计算的不断发展, 如何克服穷举算法带来的指数爆炸问题已成为 DNA 计算领域的重要研究目标之一。为减少图 3-着色问题 DNA 计算机算法中的 DNA 链数, 本文将 Adleman-Lipton 模型生物操作与粘贴模型解空间相结合的 DNA 计算模型进行扩展, 通过设计顶点着色器、稀疏图/稠密图搜索器, 提出一种用于求解图 3-着色问题的 DNA 计算模型与算法。将本算法与同类算法对比分析表明: 本算法在保持多项式操作时间的条件下, 将求解 n 个顶点的图 3-着色问题所需 DNA 分子链数从 $O(3^n)$ 减少至 $O(2^n)$, 改进了 3-着色问题同类文献的研究结果。

关键词: DNA 超级计算; 图 3-着色问题; 剪枝策略; NP 完全问题

中图分类号: TP384 **文献标识码:** A **文章编号:** 0372-2112 (2008) 11-2096-06

An $O(2^n)$ Volume Molecular Solutions for the 3-Colorable Problem on DNA-Based Supercomputing

LI Ken-li¹, ZHOU Xu¹, XU Jin²

(1. School of Computer and communications, Hunan University, Changsha, Hunan 410082, China;

2. Institute of Biomolecular Computer, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China)

Abstract: The DNA volume which increases in a pure exponentially with the scale of the problem has become the bottleneck problem. So how to decrease the volume in DNA computers is of a great significance in the research of DNA computing. For the objective to decrease the DNA volume of the 3-colorable problem, an improved DNA computing model basing on the biological operations in the Adleman-Lipton model and the solution space of stickers in the sticker-based model is introduced. Furthermore, a new DNA algorithm where new algorithms of Vertex Shader, Sparse Parallel Searcher and Dense Parallel Searcher are developed to solve the 3-colorable problem. The proposed algorithm can solve the 3-colorable problem by using the $O(2^n)$ shorter DNA strands on the condition of not varying the time complexity, as compared by far the best molecular algorithm for it in which $O(3^n)$ DNA strands is used.

Key words: DNA-based supercomputing; 3-colorable problem; pruning strategy; NP-complete problem

1 引言

图着色问题是一个著名的组合优化问题, 在理论和工程应用中具有良好的应用前景^[1]。

1994 年, Adleman 开拓性的应用 DNA 计算解决了 7 顶点有向 Hamilton 路径问题^[2]。2002 年, Adleman 小组在一台半自动化的装置上成功的解决了一个 20 个变量的可满足问题, 这是迄今为止 DNA 生物分子实验解决的规模最大的 NP 完全问题^[3]。Adleman 方法的基本思想是: 首先生成问题的所有可能解, 然后通过各种生化技术过滤掉非解, 这种方法称为生成过滤 (Generate and Filter) 模型^[4]。该模型遇到的最大障碍是“指数爆炸”问题, 是 DNA 计算目前面临的巨大挑战。为了解决“指数

爆炸”问题, 目前主要有以下方法: N. Morimoto 等^[5]提出的构造删除策略; Chen 等提出的将启发式算法转化为 DNA 计算模型的思想^[6]; 李源等在 DNA 计算中使用遗传算法的方法^[7]; 基于传统分治策略的 DNA 计算机算法设计技术^[8], 以及李大法等提出的误差模型^[9]等。

和 SAT 问题一样, 图着色问题已成为验证 DNA 计算有效性的 benchmark 问题之一, 其 DNA 计算机算法已有相当研究^[10~13]。文献[10, 11]均基于穷举算法, 所需的 DNA 链数均为 $O(3^n)$ 。E. Bach 等^[12]提出的理论 DNA 链数仅为 $O(n \cdot 89^n)$ 与 $O(2^n)$ 的算法和 Fu 等^[13]提出的理论 DNA 链数为 $O(1.345^n)$ 的算法是对着色问题可扩展算法设计的初步研究结果。然而, 由于 DNA 分子计算的特性, 上述三种算法中缩小的初始空间很难构造, 使

收稿日期: 2007-11-26; 修回日期: 2008-04-15

基金项目: 国家自然科学基金 (No. 60603053, 60503002, 60533010); 浙江省自然科学基金 (No. Y106654); 中国博士后科学基金 (No. 20060400845)

得这些算法的可行性大大降低,而且这些算法的时间或操作复杂性也大大提高^[14].2003年,Qiu等^[15]提出了一种多项式操作时间和 $O(3^n)$ 链数的算法,但该算法缺乏生物实验可行性^[15].因此,从公开出版的文献看,目前尚无解决图3-着色问题既具较低DNA空间复杂性又具生化实验可行性的DNA计算机算法.

为此,在文献[10~15]工作的基础上,本文对图3-着色问题DNA计算机算法中的可扩展性问题进行了较深入的探索,主要工作可概述为:

(1)将文献[16,17]中粘贴模型基于二进制生成解空间扩展到基于三进制;应用图灵机中的剪枝算法设计技术,对预着色库进行搜索,以缩短生物实验时间.

(2)提出一种图3-着色问题的DNA计算机新算法,本算法在不增加多项式的生化操作复杂性条件下,将所需的DNA链数从此前的 $O(3^n)$ 减少至 $O(2^n)$.

2 图3-着色问题的DNA计算模型

求解图 $G=(V, E)$ 的3-着色问题可认为是寻找从图的顶点集 $V(G)$ 到颜色集 $\{b, r, g\}$ 的一种映射:

$f:V \rightarrow \{b, r, g\}$ 满足:若 $(u, v) \in E$,则 $f(u) \neq f(v)$.

本文使用文献[16,17]中提出的DNA计算模型,该模型把Adleman-Lipton模型^[3]的生物操作与粘贴模型(Sticker model)的解空间相结合,兼具Adleman-Lipton模型和粘贴模型(Sticker model)的优点,具有随机存储能力,杂交后误解率较低,已被成功应用于子集和^[16]、因式分解^[17]等问题的DNA计算机算法设计中.

根据图3-着色问题的定义,颜色集需要3个不同的值来表示.因此,文献[16,17]中基于二进制生成粘贴模型解空间的方法已难满足本文问题的求解需求,须对其解空间进行扩展.扩展后解空间的生成基于三进制,每个顶点的颜色值对应三段不同的DNA链,即分别产生三段不同的DNA链标示红色、蓝色、绿色.令变量 x_i 表示顶点 v_i 的颜色, f 的值域为 $\{b, r, g\}$.即

$$x_i = f(v_i) = \begin{cases} b \\ r, b \text{ 蓝色}, r \text{ 红色}, g \text{ 绿色} \\ g \end{cases} \quad (1)$$

因此,在扩展后的计算模型中,DNA链状态由原模型中的0,1(即开和关)扩展为 b, r, g .

将图3-着色问题的输入编码成DNA碱基序列并生成解空间后,本文的基本生化操作均基于Adleman-Lipton模型中的DNA操作^[3,16,17]:即抽取(Extract)、合并(Merge)、检测(Detect)、复制(Amplify)、添加(Append)、丢弃(Discard)与读取(Read).

3 图3-着色问题的DNA计算机算法

为描述方便,本文仅考虑连通图的着色问题.

3.1 图3-着色问题的DNA计算机算法思想

设 $G=(V, E)$ 是一个有限无向连通图,顶点集和边集分别为 $V=\{v_1, v_2, \dots, v_n\}$ 与 $E=\{e_1, e_2, \dots, e_m\}$.基于经前节扩展后的模型,根据DNA操作的处理特性,本算法基本过程描述如下:

算法1 图3-着色问题DNA计算机算法框架

(1)生成预着色库.选取 k 个顶点穷举生成预着色库.

(2)引入剪枝技术.应用搜索器对预着色库进行筛选.

(3)生成着色库.着色器对图中未着色的 $n-k$ 个顶点着色.

(4)搜索解.根据顶点数 n 与边数 m 的关系将图分成稀疏图和稠密图,设计不同的子算法(搜索器)搜索其最终解.

以下介绍算法1中步1-4各子算法的详细设计.

3.2 基于扩展模型的图3-着色问题预着色库

选取图 G 中 k 个点穷举生成预着色库.用一个 k 位数 $x_1^i x_2^i \dots x_k^i$ 表示 k 个顶点的一种着色方案, x_m 表示图 G 中的第 m 个顶点($0 \leq m < k$), i 标示顶点的颜色($i \in \{b, r, g\}$).令 x_m^b, x_m^r, x_m^g 分别代表顶点 x_m 颜色为 b, r, g .如 $x_1^b x_2^b \dots x_k^b$ 表示所有顶点均着蓝色.下述算法用于构建图 G 的预着色库.

算法2 预着色库的DNA算法

Procedure Init(T_0, k)

(1) For $m=1$ to k

(1a) Amplify(T_0, T_1, T_2, T_3).

(1b) Append(T_1, x_m^b).

(1c) Append(T_2, x_m^r).

(1d) Append(T_3, x_m^g).

(1e) $T_0 = (T_1, T_2, T_3)$.

EndFor

EndProcedure

引理1 算法 Init(T_0, k)可并行建立预着色库.

证明:算法 Init(T_0, k)中第(1a)步将试管 T_0 中的内容复制到试管 T_1, T_2 与 T_3 中,试管 T_0 为空.第(1b)-(1d)步执行添加操作,分别添加代表 x_m 值为 b, r, g 的DNA链到试管 T_1, T_2, T_3 中的DNA链后.第(1e)步将试管 T_1, T_2 与 T_3 中的内容合并到 T_0 中. T_0 中包含 $x_m = b, x_m = r$ 与 $x_m = g$ 的DNA链.重复(1a)-(1e),最后 T_0 中包含 $3k$ 个DNA链,这些DNA链代表 $3k$ 种可能的着色方案.因此,通过算法 Init(T_0, k)可以形成前 k 个顶点的着色库.

算法 Init(T_0, k)需要执行 k 次复制操作, $3 \times k$ 次添加操作, k 次合并操作,需要的试管数为4.假定序列每

一位的颜色用长度为 15 的 DNA 链来表示,那么实现算法 $\text{Init}(T_0, k)$ 后 DNA 链长为 $15 \times k$.

3.3 顶点着色器的 DNA 计算机算法

基于预着色库对图 G 中未着色的顶点进行着色. 顶点 v_i 的着色方法如下:在已着色的前 $i-1$ 个顶点中搜索出与 v_i 间存在边的顶点 v_j ;根据 x_j 的颜色值把试管 T_0 中的 DNA 链分别抽取到试管 T_1, T_3 与 T_4 中;根据 v_j 的颜色值对 v_i 进行着色.

算法 3 并行着色器

Procedure Vertex. Shader (T_0, m, nrk)

(1) for $i = k + 1$ to n

for $j = 1$ to $i-1$

Assume that there are an edge between the vertex v_i and v_j ($0 < j < i-1$)

(1 a) $T_1 = + (T_0, x_j^b)$ and $T_2 = - (T_0, x_j^b)$.

(1 b) $T_3 = + (T_2, x_j^r)$ and $T_4 = - (T_2, x_j^r)$.

(1 c) Amplify(T_1, T_5, T_6).

(1 d) Amplify(T_3, T_7, T_8).

(1 e) Amplify(T_4, T_9, T_{10}).

(1 f) Append(T_5, x_i^r).

(1 g) Append(T_6, x_i^g).

(1 h) Append(T_7, x_i^b).

(1 i) Append(T_8, x_i^g).

(1 j) Append(T_9, x_i^b).

(1 m) Append(T_{10}, x_i^r).

(1 n) $T_0 = (T_5, T_6, T_7, T_8, T_9, T_{10})$.

EndFor

EndFor

EndProcedure

引理 2 算法 Vertex. Shader ($T_0, m, n - k$) 实现对余下 nrk 个顶点并行着色.

证明:由算法 $\text{Init}(T_0, k)$ 得到预着色库,经过初步搜索后得到 T_0 . 第(1 a)-(1 b)步根据 v_j 的颜色值进行抽取操作形成试管 T_1, T_2, T_3, T_4 . T_1 中 $x_j = b$, T_2 中 $x_j = g$ 或 r , T_3 中 $x_j = r$, T_4 中 $x_j = g$. 第(1 c)步将 T_1 中的内容复制到试管 T_5, T_6 中, T_0 为空. 同理第(1 d)步形成两个试管 T_7, T_8 . 第(1 e)步复制操作形成两个试管 T_9, T_{10} . 第(1 f)步添加代表 $x_i = r$ 的 DNA 链到 T_5 中的 DNA 链后. 同理第(1 g)-(1 m)步分别添加代表 x_i 值为 r, g, b, g, b, r 的 DNA 链到试管 $T_6, T_7, T_8, T_9, T_{10}$ 中的 DNA 链后. 第(1 n)步把试管 T_5, T_6, T_7, T_8, T_9 与 T_{10} 中的内容合并到试管 T_0 中. 最后 T_0 中形成了着色库.

算法 Vertex. Shader(T_0, m, nrk) 需要执行 $2 \times (nrk)$ 次抽取操作, $3 \times (nrk)$ 次复制操作, $6 \times (nrk)$ 次添加操作, (nrk) 次合并操作, DNA 链长度为 $15 \times n$, 需要的试

管数为 10.

这样,算法 2 和 3 完成后,图 G 的着色库将告生成.

3.4 解搜索 DNA 算法

文献[16~18]中搜索过程的 DNA 算法均使用直接穷举方法,这样的搜索过程将非常耗时.受文献[15]启示,本文将图分成稀疏图和稠密图分别处理.同时利用粘贴模型的思想,在搜索时逐步剔除非合法解链.

3.4.1 稀疏图并行搜索器

若一条边的两个顶点着相同颜色,则对应的为非法 DNA 链.令图 3-着色问题解搜索器的边数 p ($0 < p < m$),以下详叙稀疏图搜索器 DNA 计算机算法的设计.

算法 4 稀疏图并行搜索器

Procedure Sparse. Parallel. Searcher(T_0, p, n)

(1) For $i = 1$ to p

Assume that the i th edge is between vertex v_{i1} and v_{i2}

(1 a) $T_1 = + (T_0, x_{i1}^b)$ and $T_2 = - (T_0, x_{i1}^b)$.

(1 b) $T_3 = + (T_2, x_{i1}^r)$ and $T_4 = - (T_2, x_{i1}^r)$.

(1 c) $T_5 = + (T_1, x_{i2}^b)$ and $T_6 = - (T_1, x_{i2}^b)$.

(1 d) $T_7 = + (T_3, x_{i2}^r)$ and $T_8 = - (T_3, x_{i2}^r)$.

(1 e) $T_9 = + (T_4, x_{i1}^g)$ and $T_{10} = - (T_4, x_{i1}^g)$.

(1 f) $T_0 = (T_6, T_8, T_{10})$.

(1 g) $T_5 = (T_5, T_7, T_9)$.

EndFor

(2) If (Detect(T_0) = 'yes') then

(2 a) Read(T_0).

EndIf

(3) If (Detect(T_5) = 'yes') then

(3 a) Discard(T_5).

EndIf

EndProcedure

引理 3 算法 Sparse. Parallel. Searcher(T_0, m, n) 实现对着色库的搜索,找到合法解链.

证明:针对稀疏图,该搜索器可根据边的两端是否着相同的颜色来对 DNA 链进行筛选.搜索过程需要考虑 p 条边,边的两端用 v_{i1}, v_{i2} 标示.其颜色值分别用 x_{i1}, x_{i2} 标示.第(1 a)-(1 e)步抽取操作得到试管 $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9$ 与 T_{10} .其中 T_1 中 $x_{i1} = b$, T_2 中 $x_{i1} = r$ 或 g , T_3 中 $x_{i1} = r$, T_4 中 $x_{i1} = g$, T_5 中 $x_{i2} = b$, T_6 中 $x_{i2} = r$ 或 g , T_7 中 $x_{i2} = r$, T_8 中 $x_{i2} = b$ 或 g , T_9 中 $x_{i2} = g$, T_{10} 中 $x_{i2} = r$ 或 b .第(1 f)步把 T_6, T_8, T_{10} 中的 DNA 链合并到 T_0 中. T_0 中包含合法解链.第(1 g)步把 T_5, T_7, T_9 中的 DNA 链合并到 T_5 中.第(2)步检测 T_0 中是否含有 DNA 链.如果返回 yes 执行第(2 a)步读取 T_0 中 DNA 链所包含的信息.第(3)步检测 T_5 中是否含有 DNA 链.如果返回 yes 执行第(3 a)步丢弃 T_5 中的 DNA.

算法 Sparse-Parallel-Searcher(T_0, p, n) 需要执行 $5 \times p$ 次抽取操作, p 次合并操作, 1 次检测操作和 1 次读取操作, 1 次丢弃操作, 需要的试管数为 10.

3.4.2 稠密图并行搜索器

两个着色相同的顶点间存在边, 则对应的为非法 DNA 链. 令图 3-着色问题解搜索器的顶点数为 n_0 ($n_0 = k$ 或 nrk , 其中 k 为常数), 以下详叙稠密图搜索器 DNA 计算机算法的设计.

算法 5 稠密图并行搜索器

Procedure Dense-Parallel-Searcher(T_0, p, n_0)

(1) For $i = 1$ to $n_0 - 1$

(1a) $T_1 = + (T_0, x_i^b)$ and $T_2 = - (T_0, x_i^b)$.

(1b) $T_3 = + (T_2, x_i^r)$ and $T_4 = - (T_2, x_i^r)$.

For $j = i + 1$ to n

(1c) $T_5 = + (T_1, x_j^b)$.

(1d) $T_6 = + (T_5, e_{ij}^1)$ and $T_7 = - (T_5, e_{ij}^0)$.

(1e) $T_1 = (T_1, T_7)$.

(1f) Discard(T_6).

(1g) $T_8 = + (T_3, x_j^r)$.

(1h) $T_9 = + (T_8, e_{ij}^1)$ and $T_{10} = - (T_8, e_{ij}^0)$.

(1i) $T_3 = (T_3, T_8)$.

(1j) Discard(T_9).

(1k) $T_{11} = + (T_4, x_j^r)$.

(1m) $T_{12} = + (T_{11}, e_{ij}^1)$ and $T_{13} = - (T_{11}, e_{ij}^0)$.

(1n) $T_4 = (T_4, T_{11})$.

(1p) Discard(T_{13}).

(1q) $T_0 = (T_6, T_9, T_{13})$.

EndFor

EndFor

(2) If (Detect(T_0) = 'yes') then

(2a) Read(T_0).

EndIf

EndProcedure

引理 4 算法 Dense-Parallel-Searcher(T_0, m, n) 可实现对稠密图着色库中 DNA 链的筛选操作.

证明: 针对稠密图, 检查着色相同的顶点间是否有边, 如果有边则为非法解. 算法中第(1a)-(1b)步抽取操作形成试管 T_1-T_4 ; T_1 中 $x_i = b$, T_2 中 $x_i = r$ 或 g , T_3 中 $x_i = r$, T_4 中 $x_i = g$. 第(1c)步抽取操作形成试管 T_5 , T_5 中 $x_j = b$. 第(1d)步根据顶点 v_i 与 v_j 间是否有边把试管 T_5 分到 T_6 或者 T_7 中. 第(1e)步合并操作, 把试管 T_1 、 T_7 中的溶液合并到 T_1 中. 第(1f)步丢弃操作, 把 T_6 中的溶液丢掉. 第(1g)-(1j)步与第(1c)-(1f)步执行相似的操作, 删除非法解链. 同理第(1k)-(1p)步与第(1x)-(1f)步执行相似的操作. 第(1q)步合并操作把

T_6 、 T_9 、 T_{13} 中的 DNA 链合并到 T_0 中, 这样试管 T_0 中包含了所有的合法解链. 第(2)步检测 T_0 是否包含 DNA 链, 如果返回“yes”, 执行第(2a)步读取 T_0 中 DNA 链所包含的信息.

算法 Dense-Parallel-Searcher(T_0, p, n_0) 需要执行 $(n_0-1) \times (3n_0+2)$ 次抽取操作, $2n_0 \times (n_0-1)$ 次合并操作, $3n_0 \times (n_0-1)/2$ 次丢弃操作, 1 次检测操作, 1 次读取操作, 需要的试管数为 13.

3.5 图 3-着色问题的 $O(2^n)$ 链数 DNA 计算机算法

设生成预着色库的 k 个顶点间的边数为 q , 则余下 nrk 个顶点间的边数为 nrq . 将实现步骤 1-4 的子算法与前叙各子算法组成一个整体, 得到基于 DNA 超级计算的图 3-着色问题求解算法.

算法 1 $O(2^n)$ 链数图 3-着色问题的 DNA 计算机算法

(1) Init(T_0, k).

(2) Parallel-Searcher(T_0, q, k).

(3) Vertex-Shader(T_0, m, nrk).

(4) Parallel-Searcher(T_0, nrq, n).

定理 1 算法 1 可并行求解图 3-着色问题.

证明: 算法第(1)步执行 Init(T_0, k), 图 G 中的 k 个顶点穷举生成预着色库. 第(2)步调用算法 Parallel-Searcher(T_0, nrq, k) 对预着色库进行搜索. 这一步根据算法 1 中 k 个顶点组成的图是稀疏图还是稠密图选择搜索器. 算法第(3)步执行算法 Vertex-Shader(T_0, m, nrk), 使用顶点着色器对未着色的 nrk 个顶点进行着色, 最终生成图 G 的着色库. 第(4)步对第(3)步生成的着色库进行搜索, 最终得到合法解链即图 3-着色问题的解.

3.6 性能分析与比较

证明: 算法 1 由 4 个子算法组成, 算法在第 1 步中执行 k 次复制操作, $3 \times k$ 次添加操作, k 次合并操作. 第 3 步执行 $2 \times (nrk)$ 次抽取操作, $3 \times (nrk)$ 次复制操作, $6 \times (nrk)$ 次添加操作, (nrk) 次合并操作. 算法在第 2 步和第 4 步将根据图的性质对搜索器进行选择.

(1) $m < n \log n$

图 G 为稀疏图^[19], 使用稀疏图并行搜索器进行解链的搜索. 需执行 $5 \times p$ 次抽取操作, p 次合并操作 ($0 < p < m$), 1 次检测操作和 1 次读取操作. 此时, 本算法执行的生物操作数为 $12n + 6nr^2k + 2$, 即 $O(n + m)$.

(2) $m \geq n \log n$

图 G 为稠密图^[19], 使用稠密图并行搜索器进行解链的搜. 需执行 $(n_0-1) \times (3n_0+2)$ 次抽取操作, $2n_0 \times (n_0-1)$ 次合并操作, $3n_0 \times (n_0-1)/2$ 次丢弃操作, 1 次检测操作, 1 次读取操作. 此时, 本算法执行的生物操作数

为 $6.5n^2 + (7.5 - 13k)n + (13k^2 - 16k - 4)$ 即 $O(n^2)$ ($n_0 = k$ 或 rk, k 为常数).

故本算法总的生物操作数为 $O(n^2)$ ($O(m) = O(n^2)$).

本算法第 1 步执行完成后将生成 3^k 条 DNA 链, 算法 2 将删除算法 1 中的非法解链. 假设余下的链数为 M ($0 < M < 3^k, k$ 为常数). 根据文献[14]中的 3-顶点着色问题的 $O(2^n)$ 算法的分析可知: 算法 3 执行完毕将生成 $M \times 2^n$ 条 DNA 链. 因此, 本算法所需的 DNA 链数为 $O(2^n)$ [18]. 最坏情况下用到的测试试管数为 13 即 $O(1)$. 由引理 1 与引理 3 的结论, 算法中所有 DNA 链的最长链长为 $15 \times (k + (rk)) = 15 \times n$ 即为 $O(n)$.

文献[10, 11]、[15]中的生物操作数为 $O(n^2)$, 测试试管数 $O(1)$, DNA 链数为 $O(3^n)$, 以现有生化实验水平, 上述三种算法理论上均仅能求解 46 个顶点的 3-着色问题[15]. 与上述算法相比, 虽然本算法的测试试管数在围界于 $O(1)$ 的情形下有一定增加, 但本算法在生物操作复杂性及链长不致增加的条件下, 所需的 DNA 链数由 $O(3^n)$ 减小至 $O(2^n)$. 因此, 本算法理论上可求解着色问题的规模将可达到约 70 个顶点 ($3^{46} \approx 2^{70}$).

应该指出, 文献[12]中 3-顶点着色问题算法的生物操作数为 $O(n^2)$, 链长为 $O(n)$, DNA 链数也可达到 $O(2^n)$. 但该算法必须通过电子计算机进行预处理以得到图的生成树, 其解空间很难构造, 因而实验可行性低[15]. 本算法基于 Chang 等提出的模型[16, 17], 无须产生生成树, 解空间更易生成, 生化实验可行性更高.

因此, 和上述同类算法相比, 本算法在空间复杂性和实验可行性等综合性能上的比较优势是明显的.

4 模拟实验结果

为验证算法的有效性, 本文采用和文献[8, 16, 17]完全相同的实验方法. 以图 G(图 1 所示) 为图 3-着色问题的实例, 给出使用本算法对此实例的模拟求解过程.

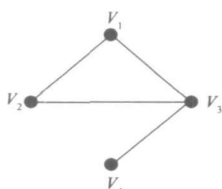


图 1 一个具有 4 个顶点的无向连通图 G

4.1 DNA 编码

本文采用 Braich 等求 20 个变量 SAT 问题中使用的 DNA 计算模型[3], 对每一变量设计两个长度均为 15 的碱基“值序列”其中表 1 标示 T_0 中 15 个变量的 DNA 序列表 1 中 x_i 为顶点的标号, x_i^b 的值用于标示图 G 中第 i 个顶点的颜色.

4.2 算法求解过程

编码完全产生后, 即可对算法进行实现. 对图 G 进行着色, 可令 $k=2$. 首先对顶点 v_1, v_2 进行着色, 得到预着色库 DNA 链 $\{x_1^b x_2^b, x_1^b x_2^r, x_1^b x_2^g, x_1^r x_2^b, x_1^r x_2^r, x_1^r x_2^g, x_1^g x_2^b, x_1^g x_2^r, x_1^g x_2^g\}$

$x_2^b, x_1^g x_2^r, x_1^g x_2^g\}$ 后对其进行搜索, 最终得到合法预解链 $\{x_1^b x_2^r, x_1^b x_2^g, x_1^r x_2^b, x_1^r x_2^g, x_1^g x_2^b, x_1^g x_2^g\}$.

表 1 T_0 中各位的 DNA 序列

位	5' 3' DNA 序列	位	5' 3' DNA 序列
x_1^b	AA TAA TTTXX TXXXX	x_1^r	XTA TX TAA TTAAAXA
x_1^g	AXX TAA XAXA XTXT	x_2^b	TTXXX AXA TXAAXX
x_2^b	XTXX TA TA TXXAXXT	x_2^g	XXA TAA TXA XTAXXX
x_3^b	TX TAT TTTX TTXAT	x_3^g	A XXXX AA XXA TTXA
x_3^g	TTXAX TA TAA TXXX	x_4^b	TTTAA TA TTXA XATT
x_4^b	XA TTX TTTX AA TAA X	x_4^g	T XXX T AX T XA T XAXA
x_5^b	T TA TTA XAA XTAA T	x_5^g	XXA XAA TAA A TTTXX
x_5^g	TTT XAA TAA XAXXTX		

基于合法预解链利用顶点着色器对余下 2 个顶点进行着色. 首先对顶点 v_3 进行着色, 找出与顶点 v_3 间存在边的顶点 v_1 , 根据顶点 v_1 的颜色把 T_0 中的 DNA 链分到 T_1, T_3, T_4 中. 试管 T_1, T_3, T_4 中顶点 v_1 颜色值分别为“b”、“r”、“g”. 对 T_1 中顶点 v_3 着颜色“r”或“g”, T_3 中顶点 v_3 着颜色“b”或“g”, 试管 T_4 中顶点 v_3 着颜色“r”或“b”. 顶点 v_3 的着色完成后得到的解链: $\{x_1^b x_2^r x_3^r, x_1^b x_2^r x_3^g, x_1^b x_2^g x_3^r, x_1^b x_2^g x_3^g, x_1^r x_2^b x_3^b, x_1^r x_2^b x_3^g, x_1^r x_2^g x_3^b, x_1^r x_2^g x_3^g, x_1^g x_2^b x_3^b, x_1^g x_2^b x_3^g, x_1^g x_2^g x_3^b, x_1^g x_2^g x_3^g\}$. 顶点 v_4 的着色过程与顶点 v_3 相似, 最后得到图 G_1 的着色库: $\{x_1^b x_2^r x_3^r x_4^b, x_1^b x_2^r x_3^r x_4^g, x_1^b x_2^r x_3^g x_4^b, x_1^b x_2^r x_3^g x_4^g, x_1^b x_2^g x_3^b x_4^b, x_1^b x_2^g x_3^b x_4^g, x_1^b x_2^g x_3^g x_4^b, x_1^b x_2^g x_3^g x_4^g, x_1^r x_2^b x_3^b x_4^b, x_1^r x_2^b x_3^b x_4^g, x_1^r x_2^b x_3^g x_4^b, x_1^r x_2^b x_3^g x_4^g, x_1^r x_2^g x_3^b x_4^b, x_1^r x_2^g x_3^b x_4^g, x_1^r x_2^g x_3^g x_4^b, x_1^r x_2^g x_3^g x_4^g, x_1^g x_2^b x_3^b x_4^b, x_1^g x_2^b x_3^b x_4^g, x_1^g x_2^b x_3^g x_4^b, x_1^g x_2^b x_3^g x_4^g, x_1^g x_2^g x_3^b x_4^b, x_1^g x_2^g x_3^b x_4^g, x_1^g x_2^g x_3^g x_4^b, x_1^g x_2^g x_3^g x_4^g\}$.

着色库生成后需对其中的 DNA 链进行筛选. 由于图 G 为稀疏图, 找出着色库生成过程中没有考虑的边 $v_2 v_3$. 顶点 v_2 与 v_3 间存在边, 删除顶点 v_2, v_3 颜色相同的 DNA 链. 首先按照 x_2 的值对 T_0 中的 DNA 链进行抽取操作得到试管 T_1, T_3, T_4 . 试管 T_1, T_3, T_4 中顶点 v_2 颜色分别为“b”、“r”、“g”. 对 T_1 中的 DNA 链进行抽取操作得到数据池 T_5 与 T_6 . T_5 中 DNA 链顶点 v_3 颜色值为“b”与顶点 v_2 着相同颜色, 由于顶点 v_2 与顶点 v_3 间存在边, 所以 T_5 中为非法解链, 对其执行丢弃操作. 数据池 T_6 中为合法解链. 按照同样的方法对数据池 T_3, T_4 中的 DNA 链进行筛选, 最终得到合法解链为: $\{x_1^b x_2^r x_3^g x_4^b, x_1^b x_2^r x_3^g x_4^g, x_1^b x_2^g x_3^b x_4^b, x_1^b x_2^g x_3^b x_4^g, x_1^b x_2^g x_3^g x_4^b, x_1^b x_2^g x_3^g x_4^g, x_1^r x_2^b x_3^b x_4^b, x_1^r x_2^b x_3^b x_4^g, x_1^r x_2^b x_3^g x_4^b, x_1^r x_2^b x_3^g x_4^g, x_1^r x_2^g x_3^b x_4^b, x_1^r x_2^g x_3^b x_4^g, x_1^r x_2^g x_3^g x_4^b, x_1^r x_2^g x_3^g x_4^g, x_1^g x_2^b x_3^b x_4^b, x_1^g x_2^b x_3^b x_4^g, x_1^g x_2^b x_3^g x_4^b, x_1^g x_2^b x_3^g x_4^g, x_1^g x_2^g x_3^b x_4^b, x_1^g x_2^g x_3^b x_4^g, x_1^g x_2^g x_3^g x_4^b, x_1^g x_2^g x_3^g x_4^g\}$.

5 结论

为解决 DNA 计算中穷举方法导致的指数爆炸问

题.在文献[10,11,18,19]工作的基础上,本文对 DNA 计算的可扩展性问题进行了较深入的探索:根据现有 DNA 计算模型的生物操作特性和 NP 完全的图 3-着色问题的并行求解需求,对文献[18,19]的 DNA 计算模型进行了扩展,基于扩展模型设计了一种求解图 3-着色问题的 DNA 计算机新算法.新算法在不提高算法生物操作复杂性的条件下,将求解图 3-着色问题的 DNA 链数由穷举方法的 $O(3^n)$ 减少至 $O(2^n)$.值得指出的是,只需将非连通图分成多个子连通图,本算法便可用于非连通图着色问题.

参考文献:

- [1] J A Bondy, U S R Murty. Graph Theory with Application, the Macmillan Press LTD [M]. London: Basingtoke and New York, 1976.
- [2] L Adleman. Molecular computation of solutions to combinatorial problems [J]. Science, 1994, 266(5187): 1021 - 1024.
- [3] R S Braich, et al. Solution of a 20-Variable 3-SAT Problem on a DNA Computer [J]. Science, 2002, 296: 499 - 502.
- [4] 李汪根, 丁永生. DNA 计算机中队列数据结构的设计及实现 [J]. 计算机学报, 2007, 6(30): 993 - 998.
LI Wang-gen, DING Yong-sheng. Design and implementation of queue data structure in DNA computer [J]. Chinese Journal Of Computers. 2007, 6(30): 993 - 998. (in Chinese)
- [5] N Morimoto, et al. Solid-phase DNA solution to the hamiltonian path problem [A]. DIMACS, Series in Discrete Mathematics and Theoretical Computer Science [C]. New York: AMS, 1999. 193 - 206.
- [6] K Chen, et al. A Space-Efficient Randomized DNA Algorithm for k-SAT [A]. The 6th International Workshop on DNA-Based Computers, Lecture Notes in Computer Science [C]. Leiden: Springer, 2000. 199 - 208.
- [7] 李源, 方辰, 等. 最大团问题的 DNA 计算机进化算法 [J]. 科学通报, 2004, 49(5): 439 - 443.
LI Yan, FANG Chen, et al. Genetic algorithm in DNA Computing: A solution to the maximal clique problem [J]. Chinese Science Bulletin, 2004. 49(9): 967 - 971. (in Chinese)
- [8] 李肯立, 姚凤娟, 等. 基于分治的背包问题 DNA 计算机算法 [J]. 计算机研究与发展, 2007, 44(6): 1063 - 1070.
LI Ken-li, YAO Feng-juan, et al. Improved molecular solution for the knapsack problem on DNA-Based supercomputing [J]. Journal of Computer Research and Development, 2007, 44(6): 1063 - 1070. (in Chinese)
- [9] D F Li, et al. Scalability of the surface-based DNA algorithm for 3-SAT [J]. BioSystems, 2006, 85: 95 - 98.
- [10] 许进, 强小利, 等. 一种图顶点着色 DNA 计算机模型 [J]. 科学通报, 2006, 51(4): 480 - 487.
XU Jin, QIANG Xiao-li, et al. A DNA computer model for solving vertex coloring problem [J]. Chinese Science Bulletin, 2006, 51(20): 2541 - 2549. (in Chinese)
- [11] 高琳, 许进. 图的顶点着色问题的 DNA 算法 [J]. 电子学报, 2003, 31(4): 494 - 497.
GAO Lin, XU Jin. A DNA algorithm for graph vertex coloring problem [J]. Acta Electronica Sinica, 2003, 31(4): 494 - 497. (in Chinese)
- [12] E Bach, et al. DNA models and algorithms for NP-complete problems [J]. Journal of Computer and System Science, 1998: 172 - 186.
- [13] B Fu. Volume Bounded Molecular Computation [D]. Department of Computer Science, Yale University, 1997.
- [14] N C Leandro. Fundamentals of natural computing: an overview [J]. Physics of Life Reviews, 2007, 4: 1 - 36.
- [15] Z F Qiu, M Lu. A new approach to advance the DNA computing [J]. Applied Soft Computing, 2003(3): 177 - 189.
- [16] W L Chang, et al. Molecular solutions for the subset-sum problem on DNA-based supercomputing [J]. BioSystems, 2004, 73: 117 - 130.
- [17] W L Chang, et al. Fast parallel molecular solutions for DNA-based supercomputing: Factoring Integers [J]. IEEE Transactions on Nanobioscience, 2005, 4(2): 149 - 163.
- [18] K H Zimmermann. Efficient DNA sticker algorithms for NP-complete graph problems [J]. Computer Physics Communications, 2002, 144: 297 - 309.
- [19] M C Golumbic. Algorithmic Graph Theory and Perfect Graphs [M]. Academic Press, New York, 1980.

作者简介:



李肯立 男, 1971 年生于湖南娄底, 教授, 中国计算机学会高级会员, 研究方向为并行处理、生物计算、网格计算等。
E-mail: lk1510@263.net



周旭 女, 1983 年生于江苏宿迁, 硕士研究生, 研究方向为并行计算、DNA 计算。
E-mail: happypana2006@126.com

许进 男, 1959 年生于陕西乾县, 华中科技大学教授, 博士生导师, 主要研究方向为 DNA 计算、DNA 计算机等。