

# 子集和问题的 $O(1.414^n)$ 链数 DNA 计算机算法

李肯立<sup>1),2)</sup> 姚凤娟<sup>1)</sup> 许 进<sup>2)</sup> 李仁发<sup>1)</sup>

<sup>1)</sup>(湖南大学计算机与通信学院 长沙 410082)

<sup>2)</sup>(华中科技大学分子生物计算机研究所 武汉 430074)

**摘 要** 随着 DNA 计算机研究的不断深入,如何克服 DNA 生物计算中穷举法的极限已成为 DNA 计算研究的重要内容之一. 为设计可扩展的子集和问题 DNA 计算机算法,文中将 Aldeman-Lipton 模型的操作与粘贴模型的解空间结合,引入荧光标记和凝胶电泳技术,通过设计 DNA 并行搜索器,提出一种求解子集和问题的 DNA 计算机模型和算法. 与已有文献结论的对比分析表明:文中算法在保持多项式生物操作复杂性的条件下,将穷举算法中的 DNA 分子链数从  $O(2^n)$  减少至  $O(1.414^n)$ ,其中  $n$  为子集和问题的维数. 因此,文中算法理论上在试管级生化反应条件下能将可破解子集和公钥的维数从 60 提高到 120.

**关键词** DNA 计算;子集和问题;分治法;并行处理;NP 完全问题

**中图法分类号** TP301

## An $O(1.414^n)$ Volume Molecular Solutions for the Subset-Sum Problem on DNA-Based Supercomputing

LI Ken-Li<sup>1),2)</sup> YAO Feng-Juan<sup>1)</sup> XU Jin<sup>2)</sup> LI Ren-Fa<sup>1)</sup>

<sup>1)</sup>(School of Computer and Communication, Hunan University, Changsha 410082)

<sup>2)</sup>(Institute of Biomolecular Computer, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** How to decrease the volumes in DNA computers has become an important research area in DNA computing. It is showed that the poor scalability in the DNA-based algorithms roots in the poor scalability of the DNA models. In this paper, a DNA model for good scalability is proposed. It is based on biological operations in the Adleman-Lipton model and the solution space of stickers in the sticker-based model. The method of fluorescence labeling and the technique of gel electrophoresis are incorporated into the model. Based on it, a new DNA algorithm for solution of the subset-sum problem is proposed where the strategy of divide and conquer and a new designed algorithm of ParallelSearcher are introduced. The proposed algorithm can solve the  $n$ -dimension subset-sum instances by using the  $O(1.414^n)$  shorter DNA strands on the condition of not varying the time complexity, as compared by far the best molecular algorithm for it in which  $O(2^n)$  DNA strands is used. Therefore, the scale of the public key cryptosystem that can be theoretically broken using the present biological technology may be enlarged from 60 to 120 variables.

**Keywords** DNA-based computing; subset-sum problem; divide and conquer; parallel processing; NP-complete problem

## 1 引言

DNA 计算是一种以 DNA 分子和相关生物酶等作为基本材料,以某些生化反应为基础的一种新的计算模式. 1994 年,Adleman 开创性地使用基于 DNA 分子的生化反应解决了 7 个顶点的有向 Hamilton 路径问题<sup>[1]</sup>,此后,关于 DNA 计算机及其计算模型与实验方法等方面的研究日益引起重视<sup>[2]</sup>,诸多学者相继给出了不同类型的图与组合优化问题的 DNA 计算机算法和实验结果<sup>[3-6]</sup>.

目前为止,一个测试试管已可产生  $10^{18}$  个 DNA 链,它可使  $10^{18}$  位数据以数据并行的方式并行运行<sup>[7]</sup>,即 DNA 计算机可提供相当于  $10^{18}$  个处理单元的并行性和  $O(10^{18})$  的存储空间. DNA 计算的本质是将传统计算机求解 NP 完全问题的时间代价转换为 DNA 计算的空间即生物分子数目代价,因此,在基于穷举的 DNA 计算机算法中,图灵机算法中纯指数阶的时间转化成了 DNA 计算机算法中的纯指数量级的 DNA 分子链数,这是目前求解 NP 完全问题的 DNA 计算机算法中 DNA 链数为纯指数阶的原因.

另一方面,通过许多计算机科学家多年的努力,多数 NP 完全问题都已发现比蛮力搜索方法更好的亚指数时间算法,如子集和问题、精确的可满足性问题、集覆盖问题等背包类 NP 完全问题和三色问题以及独立集问题等<sup>[8-9]</sup>. 如能设计出和这些亚指数时间算法相应的 DNA 计算机算法,将大大减少求解这些问题的 DNA 计算机算法中的 DNA 链数,从而扩大 DNA 计算机解决难解问题的规模,这方面已有不少研究. Bach 等<sup>[10]</sup>提出在理论上实现 DNA 链数仅为  $O(n1.89^n)$  的 3 色问题的 DNA 算法以及链数为  $O(1.67^n)$  的独立集问题 DNA 计算机算法. Fu 等<sup>[11]</sup>提出链数为  $O(1.497^n)$  求解 3-SAT 问题的算法、链数  $O(1.345^n)$  3 色问题算法和链数为  $O(1.229^n)$  的独立集问题算法. 但由于缩小的 DNA 分子初始空间很难构造,同时这些算法操作复杂性大大提高,使得它们的可行性极大地降低<sup>[12]</sup>. Li 等<sup>[5]</sup>将进化算法首次引入最大团问题的 DNA 计算中,提出一种求解最大团问题的 DNA 计算机概率算法,该算法可在不显著增加 DNA 计算时间的前提下,大大减少直接穷举方法试管中 DNA 分子数目. 但这种方法的成功仅限于问题规模不太大的个例, DNA 计算机算法的低可扩展性问题仍然普遍存在.

因此,设计求解上述经典 NP 完全问题亚指数 DNA 链数的 DNA 计算机算法,目前仍存在实质困难. 近年来的研究表明<sup>[12]</sup>: DNA 计算模型和 DNA 计算机穷举算法密切相关,求解 NP 问题算法上的难以扩展来源于 DNA 计算模型上的低扩展性. 鉴此,本文以子集和问题为实例,对具有较好可扩展性的 DNA 计算机模型和算法进行了一定的探索,主要工作为:

(1) 为使求解子集和问题的 DNA 计算机算法具有较好的扩展性,将 Aldeman-Lipton 模型的操作与粘贴模型的解空间将结合,同时在 DNA 生物计算中引入荧光标记和凝胶电泳技术,设计了一种子集和问题的 DNA 计算机模型.

(2) 利用分治策略,提出一种基于新模型的子集和问题的 DNA 计算机算法,和文献[13-14]的两种算法相比,本算法并不增加操作复杂性,却将算法所需的 DNA 链数从纯指数的  $O(2^n)$  减少至亚指数的  $O(1.414^n)$ .

因此,理论上,本算法在试管级水平上可将利用 DNA 分子计算破解子集和公钥的维数从文献[13-14]的 60 提高到 120.

本文工作表明:分治法等传统算法设计技术在具有高并行度和高密度存储能力的 DNA 计算机算法设计中仍是适应的;如果未来 DNA 计算机技术日趋成熟,则基于子集和问题或大数因数分解等难问题的公钥系统将不再安全.

## 2 二表算法及 DNA 计算模型

子集和问题(也称背包问题)可描述如下:给定  $n$  个正整数的集合  $W = \{w_1, w_2, \dots, w_n\}$  和正整数  $M$ ,要求决定二值  $n$  元组  $X = (x_1, x_2, \dots, x_n)$  的值,使得式  $\sum_{i=1}^n w_i x_i = M$  被满足. 子集和问题属于经典的 NP 完全问题,除非  $NP = P$ ,否则该问题不可能存在多项式时间算法. 由于求解该问题的指数复杂性,该问题在信息密码学领域和数论研究中具有极重要的应用<sup>[15]</sup>.

2003 年, Yin 等提出一种基于表面模型求解 0-1 规划的 DNA 计算机算法<sup>[14]</sup>,由于子集和问题可视为一类特殊的 0-1 规划问题,因此该算法求解子集和问题的 DNA 链数和生物操作复杂性分别为  $O(2^n)$  和  $O(n)$ <sup>[14]</sup>. 最近, Chang 等提出了链数和操作数分别为  $O(2^n)$  和  $O(qn)$  的子集和问题 DNA 计

算机算法<sup>[13]</sup>, 其中  $q$  为  $M$  的二进制位数, 上述算法均属于简单穷举算法, 由于目前的生物技术只能使 DNA 的浓度达到微摩尔量级, 因此, 这些算法理论上在试管级生化反应基础上可破解的背包密钥的维数仅为  $60(2^{60} \approx 10^{18})$ . 但最新并行算法已可在相应计算平台上破解 80 维的背包密钥<sup>[16]</sup>. 因此, 为发挥 DNA 计算机较传统计算模型所具有的超级并行运算的优势, 解决 DNA 计算存在的指数爆炸问题, 必须设计和研究链数更少同时又保持多项式操作次数的新的 DNA 计算机算法.

## 2.1 二表算法

分治方法的引入被认为是求解 SAT 以及最大团问题等背包类 NP 完全问题算法设计最成功的策略之一. 1974 年, Horowitz 和 Sahni<sup>[8]</sup> 提出了基于分治方法的著名二表算法. 二表算法的主要贡献是将求解子集和问题的时间复杂性从穷举法或分枝限界算法的  $O(2^n)$  降低到  $O(2^{\frac{n}{2}})$ , 即  $O(\sqrt{2}^n) \approx O(1.414^n)$ <sup>[8]</sup>, 迄今为止, 该算法仍是求解背包类 NP 完全问题最有效的方法之一. 将二表算法的操作用 DNA 分子生物操作实现, 初步研究发现, 单纯将 DNA 链数大大降低的目标确可实现: 在文献[17]中, 将二表算法进行修改, 然后用 DNA 分子操作实现, 可将基于穷举的 DNA 算法的  $O(2^n)$  DNA 链数减少至  $O(1.414^n)$ . 然而, 简单的生物操作转换, 二表算法中的二表搜索阶段却无法利用 DNA 计算的内在并行性, 在将算法所需的 DNA 链从文献[13]的  $O(2^n)$  减少至  $O(1.414^n)$  的同时, 其生物操作复杂性也上升为伪多项式的  $O((q+n)M)$ , 其中  $q$  为  $M$  的二进制位数<sup>[13]</sup>. 显然, 伪多项式的时间复杂性仍是限制 DNA 计算机算法走向应用的重要因素.

这样, 一个自然的问题是: 根据 DNA 分子操作的并行特性, 在 DNA 分子算法设计中引入分治策略, 可否达到既减少算法中的 DNA 链数、又不致使 DNA 分子操作次数从多项式增加到伪多项式的目标? 由于现有 DNA 计算模型在二表搜索时无法发挥 DNA 计算的内在并行性<sup>[17]</sup>, 即算法的扩展性过低源于 DNA 模型的可扩展性过差. 因此, 必须从 DNA 计算模型着手, 使 DNA 计算机在二表搜索过程的内在并行性得到实现.

## 2.2 子集和问题 DNA 计算模型

与文献[4, 6, 13]的方法类似, 本文采用 Aldeman-Lipton 模型中的生物操作, 其解空间是具有随机访

问内存的粘贴模型的解空间. 该模型是一种通用的 DNA 计算模型, 已成功求解子集和<sup>[13]</sup>、子集积<sup>[6]</sup>、大数因式分解<sup>[4]</sup>等难解问题, 实现了加、减、乘、除这些复杂的算术运算.

Aldeman-Lipton 模型中 DNA 操作如下: 抽取 (Extract) ( $+(P, S)$  表示试管  $P$  中所有包含  $S$  作为子链的 DNA 分子;  $-(P, S)$  表示  $P$  中所有不包含  $S$  作为子链的 DNA 分子)、合并 (Merge) (记作  $\cup(P_1, P_2)$ , 其中  $P_1$  和  $P_2$  为试管)、检测 (Detect)、复制 (Amplify)、添加 (Append)、读取 (Read).

2004 年, 为在理论上进一步完善 DNA 计算粘贴模型, 许进等<sup>[18]</sup> 拓展了粘贴模型的应用范围, 提出了两种扩展的粘贴模型:  $k$ -进制粘贴模型和全信息粘贴 DNA 计算模型, 并指出将荧光技术程序化地加入粘贴模型, 将使此模型进一步通用化. 为实现引入分治法后 DNA 计算的内在并行性, 在二表算法的 DNA 算法设计上作如下转换: 搜索算法中将二进制运算结果转换成等长的 DNA 链. 同时为使 DNA 计算机适应这一转换, 受这种对粘贴模型进行扩展思想的启示, 在 Aldeman-Lipton 模型原有 6 种基本生物操作的基础上, 增加下述两种生物操作: (7) 荧光标记; (8) 凝胶电泳. 值得指出的是: 上述两种操作均为 DNA 计算中的常用技术, 在生化实验上不仅可行, 而且简单. 因此, 从生化实验可行性分析, 增加的生物操作后的模型并不增加生化实验的难度.

## 3 子集和问题的 DNA 计算机算法

### 3.1 基于分治的 DNA 计算机算法思想

利用分治策略, 基于所提出的模型, 根据 DNA 操作的处理特性, 本算法用 DNA 分子操作实现的基本过程描述如下.

**算法 1.** 子集和问题的 DNA 计算机算法框架.

1. 分别在  $T_{01}$  和  $T_{02}$  中用 DNA 链生成  $W_1 = \{w_1, w_2, \dots, w_{n/2}\}$  和  $W_2 = \{w_{n/2+1}, w_{n/2+2}, \dots, w_n\}$  所有子集. 对于试管  $T_{01}$  中每个不同的 DNA 链的前  $n/2$  个不同的子段分别连接上同一种颜色的荧光素, 在  $T_{02}$  中做类似操作 (与  $T_{01}$  中荧光素颜色不同) 的荧光素.

2. 分别在  $T_{01}$  和  $T_{02}$  中生成  $W_1$  和  $W_2$  的所有子集的对

应元素.

3. 在试管  $T_M$  中, 将  $M$  表示成 DNA 分子链.

4. 用并行减法器对  $T_{01}$  做基于 DNA 操作的减法, 得到  $M$  与  $T_{01}$  中各子集和之差的 DNA 链;  $T_{02}$  做 DNA 并行加法

器运算,生成  $W_2$  所有子集的子集和。

5. 使用荧光标记和凝胶电泳技术,利用并行搜索器对  $n$  位差信息与和信息进行搜索,判断问题是否有解,找解。

算法 1 中步 1~4 中的子集及其元素的 DNA 链表示算法、数  $M$  的 DNA 链表示算法以及并行加法均与文献[13]中的相应算法  $Init()$ 、 $Value()$ 、 $MakeValue()$  和  $Parallel\ Adder()$  类似,步 4 中的并行减法器与文献[4]中的相应算法  $Parallel\ Substracter()$  类似,限于篇幅,本文不再赘叙。以下介绍算法 1 的步 5 中 DNA 子算法的详细设计。

### 3.2 并行搜索器

和文献[13]的记法相同,本算法中,试管  $T_{02}$  中的  $y_{n/2 \times q+j}$  表示子集和的第  $j$  位。若某子集和的第  $j$  位为 1,则用  $y_{n/2 \times q+j}^1$  标记;否则用  $y_{n/2 \times q+j}^0$  标记。试管  $T_{01}$  中的  $u_{n/2 \times q+j}$  表示数  $M$  与试管  $T_{01}$  中各子集和做减法运算后的差值的第  $j$  位。 $u_{n/2 \times q+j}$  的标记方法与  $y_{n/2 \times q+j}$  相同(其中  $q$  为数  $M$  的二进制位数)。

#### 算法 2. 并行搜索器。

Procedure  $ParallelSearcher(T_{01}, T_{02})$

1. For  $j=1$  to  $q$

1.1.  $T_1 = +(T_{01}, u_{n/2 \times q+j}^1)$  and  $T_2 = -(T_{01}, u_{n/2 \times q+j}^1)$ .

1.2.  $T_3 = +(T_{02}, y_{n/2 \times q+j}^1)$  and  $T_4 = -(T_{02}, y_{n/2 \times q+j}^1)$ .

1.3. Append a single-stranded DNA with the length of  $2^{j-1}$  onto every strand in  $T_1$ .

1.4. Append a single-stranded DNA with the length of  $2^{j-1}$  onto every strand in  $T_3$ .

1.5.  $T_{01} = \cup(T_1, T_2)$  and  $T_{02} = \cup(T_3, T_4)$ .

EndFor

2.  $T_{01} = \cup(T_{01}, T_{02})$ .

3. 使用凝胶电泳技术将  $T_{01}$  中的分子按照链长进行分离,通过激光共焦距显微镜观察,将链长相等且具有两种颜色的 DNA 链分离,其对应的子集的并集即为子集和问题的解。

EndProcedure.

**引理 1.** 算法  $ParallelSearcher(T_{01}, T_{02})$  可并行搜索  $M$  与  $T_{01}$  中各 DNA 链表示的子集和之差和  $T_{02}$  中各 DNA 链表示的子集和相等的链。

证明. 算法  $ParallelSearcher(T_{01}, T_{02})$  中第 1.1 步使用抽取操作,将  $T_{01}$  中差值  $u_{n/2 \times q+j}$  的第  $j$  位按照 0 与 1 分离。 $T_1$  中  $u_{n/2 \times q+j} = 1$ ,  $T_2$  中  $u_{n/2 \times q+j} = 0$ 。第 1.2 步使用抽取操作,将  $T_{02}$  中和值  $y_{n/2 \times q+j}$  的第  $j$  位按照 0 与 1 分离。 $T_3$  中  $y_{n/2 \times q+j} = 1$ ,  $T_4$  中  $y_{n/2 \times q+j} = 0$ 。步 1.3 和步 1.4 分别使用添加操作,将一段链长为  $2^j$  的 DNA 链分别添加到试管  $T_1$  和  $T_3$  的每条链的末尾。步 1.5 使用合并操作将  $T_1$  与  $T_2$  合并到  $T_{01}$ ,  $T_3$  与  $T_4$  合并到  $T_{02}$ 。当循环结束时,试

管  $T_{01}$  中的差与  $T_{02}$  中的和即由二进制转换成等长的 DNA 链。

算法第 2 步将试管  $T_{01}$  与  $T_{02}$  合并到  $T_{01}$ 。第 3 步使用凝胶电泳技术将  $T_{01}$  中的分子按照链长大小进行分离,并通过激光共焦距显微镜观察链长相等的 DNA 链中是否存在两种颜色,若存在,则其对应的子集即为子集和问题的解。

算法  $ParallelSearcher(T_{01}, T_{02})$  中,共使用  $2q$  次抽取、 $2q$  次添加、 $2q+1$  次合并操作完成并行搜索。算法在执行加法器和减法器前 4 步后,其链长为  $O(qn)$ (参见文献[4,13]),因此,执行完  $ParallelSearcher(T_{01}, T_{02})$  后,链长为  $O(qn+M)$ ,使用的试管数为 6。

证毕。

### 3.3 子集和问题 $O(1.414^n)$ 链数的 DNA 计算机算法

将实现步 1~4 的子算法(与文献[4,13]的相应算法类似)与前叙各子算法组成一个整体,得到基于 DNA 超级计算的子集和问题求解算法。

**算法 3.**  $O(1.414^n)$  链数求解子集和问题的 DNA 计算机算法。

1.  $Init(T_{01}, n/2)$  and  $Init(T_{02}, n/2)$ .

2.  $Value(T_{01}, n/2, q)$  and  $Value(T_{02}, n/2, q)$ .

3.  $MakeValue(T_M, n/2, q)$ .

4.  $ParallelSubtracter(T_{01}, n/2, q, j)$  and

$ParallelAdder(T_{02}, n/2, q)$ .

5.  $ParallelSearcher(T_{01}, T_{02})$ .

**定理 1.** 算法 3 可并行求解子集和问题。

证明. 算法第 1 步  $Init(T_{01}, n/2)$  和  $Init(T_{02}, n/2)$  将集合  $W_1$  和  $W_2$  对应的子集用 DNA 链表示,并通过两种不同的荧光素表示来自不同集合的子集。第 2 步  $Value(T_{01}, n/2, q)$  和  $Value(T_{02}, n/2, q)$  将集合  $W_1$  和  $W_2$  的子集中的元素用 DNA 链表示。第 3 步  $MakeValue(T_M, n/2, q)$  将正整数  $M$  用 DNA 链表示(此 3 步的算法参见文献[13])。第 4 步  $ParallelSubtracter(T_{01}, n/2, q, j)$  求解  $M$  与  $T_{01}$  中  $W_1$  各子集和的差(参见文献[4]),  $ParallelAdder(T_{02}, n/2, q)$  求解  $W_2$  中各子集和(参见文献[13])。第 5 步  $ParallelSearcher(T_{01}, T_{02})$  并行搜索  $M$  与  $T_{01}$  中各 DNA 链表示的子集和之差与  $T_{02}$  中各 DNA 链表示的子集和相等的链,其 DNA 链对应的子集即子集和问题的解。

证毕。

### 3.4 性能分析与比较

**定理 2.** 令数  $M$  的二进制位数为  $q$ , 算法 3 求解  $n$  维子集和问题所用到的生物操作数为  $O(qn)$ , 最大链长  $O(qn+M)$  度测试试管数为  $O(1)$ , DNA

链数为  $O(1.414^n)$ 。

证明. 算法 1 由 5 个子算法组成. 前 4 步中使用的子算法与文献[4,13]的相应算法类似,由文献[4,13]的结论易知:本算法第 1 步使用  $n$  个复制、 $2n$  个添加、 $n$  个合并操作. 第 2 步使用  $2 \times n \times q$  个添加、 $n$  个合并和  $n$  个抽取操作. 第 3 步使用  $q$  个添加操作. 第 4 步使用  $2q + n + 16 \times q \times n$  个添加、 $n \times q$  个合并和  $7 \times n \times q$  个抽取操作. 算法第 5 步使用  $2q$  次抽取、 $2q$  次添加、 $2q + 1$  次合并操作. 故生物操作数为  $O(qn)$ . 由引理 1 可知,本算法中 DNA 链的最大链长为  $O(qn + M)$ 。

由文献[4,13]中的算法分析知:本算法用到的测试试管数为 16,即  $O(1)$ . 算法利用分治策略将包含  $n$  个元素的集合  $W$  平均分成两个各包含  $n/2$  个元素的集合  $W_1$  和  $W_2$ ,在子算法  $Init(T_{01}, n/2)$  和  $Init(T_{02}, n/2)$  中各生成  $2^{n/2}$  个 DNA 链,此后算法在执行过程中没有再生成新的 DNA 链,因此,总的 DNA 链数为  $O(2^{n/2})$  即  $O(\sqrt{2}^n) \approx O(1.414^n)$ . 证毕.

文献[14]的 DNA 计算机算法求解子集和问题时的 DNA 链数为纯指数的  $O(2^n)$ ,操作复杂性为  $O(n)$ ,链长为  $O(M)$ . 文献[13]中求解  $n$  维子集和问题算法中的 DNA 链数为  $O(2^n)$ ,操作复杂性为  $O(qn)$ ,链长为  $O(qn)$ . 而文献[17]中基于分治策略求解子集和问题的 DNA 计算机算法中,算法的 DNA 链数虽首次达到亚指数的  $O(1.414^n)$ ,但操作复杂性却上升为伪多项式的  $O((q + M)n)$ ,链长为  $O(qn)$ . 本文所提出的子集和问题 DNA 计算模型求解  $n$  维子集和问题的 DNA 计算机算法中,不仅将 DNA 链数降低至亚指数的  $O(1.414^n)$ ,而且保持了时间复杂性仍为多项式的  $O(qn)$ 。

值得指出的是,本算法中 DNA 链的最大链长  $O(qn + M)$  仍为伪多项式阶,但注意到目前涉及求和及带权图等 NP 问题的 DNA 算法中,即使是基于穷举的 DNA 计算机算法,也是简单通过添加相应链长的分量及权值来实现的,其 DNA 链长亦为伪多项式阶. 另外,由于目前生化反应的 DNA 链长可达数千万碱基,而实验中的单次生物操作却需要较长的时间<sup>[1]</sup>. 因此,本算法和上述算法的比较优势仍是明显的。

### 4 算法实现

为验证本文算法的有效性,以  $W = \{1, 2, 3, 4\}$ ,  $M = 5$  作为子集和问题实例. 以下给出使用本算法

对此实例的模拟求解过程.

#### 4.1 DNA 编码

本文编码采用 Braich 等求 20 个变量 SAT 问题中使用编码规则<sup>[3]</sup>,对每一变量设计两个长度均为 15 的碱基“值序列”,分别用来表示该变量的二进制 0 和 1. 依据 Braich 的编码规则,在 Windows XP 操作系统下使用 Visual C++ 6.0 的编译器来产生 DNA 序列. 表 1 是  $T_{01}$  中 25 个变量的 DNA 序列. 为实现最终解搜索的并行性,在初始生成  $W_1$  和  $W_2$  所有子集的过程中,采用荧光标记方法,在试管  $T_{01}$  中每个不同的 DNA 链的前  $n/2$  个不同子段分别连接上红色的荧光素,对试管  $T_{02}$  中的 DNA 链做同样的操作,注意连接绿色的荧光素.  $T_{02}$  中编码方法同  $T_{01}$ .

表 1 中  $x_i$  用于标示集合中的第  $i$  个元素是否在该子集中出现<sup>[13]</sup>;  $s_{i,j}$  用作第  $i$  元素转换成二进制后的第  $j$  位值<sup>[13]</sup>.  $l_i$  为减法运算中第  $i - 1$  次运算产生的借位.

表 1  $T_{01}$  中 25 个变量的 DNA 序列

位	5'→3' DNA 序列	位	5'→3' DNA 序列
$x_1^0$	AATTCACAAACAATT	$x_1^1$	AATTCACAAACAATT
$x_2^0$	AATTCACAAACAATT	$x_2^1$	AAATTAATACATTAA
$s_{1,1}^0$	AATTTCCCATTCCCTA	$s_{1,1}^1$	AATTCATCATCAATT
$s_{1,2}^0$	TCTCTCTCTAATCAT	$s_{1,2}^1$	CCATCATCTACCTTA
$s_{1,3}^0$	CTTCTCCACTATACT	$s_{1,3}^1$	CCTAAATCTCCAATA
$s_{2,1}^0$	CCTTTCTAACCCTTCA	$s_{2,1}^1$	TATCTTTCTTTATCA
$s_{2,2}^0$	AAACTCTACATACAC	$s_{2,2}^1$	TTTACCCTCATTACT
$s_{2,3}^0$	AATTAACAATCATCT	$s_{2,3}^1$	AATTCACTTTCTATC
$u_1^0$	TTACTCTTAACATCT	$u_1^1$	CCACCCTCATCCTAT
$u_2^0$	TTAATCAAAATCCCTA	$u_2^1$	CTCTTAATCTCATTC
$u_3^0$	ATTCTAACTCTACCT	$u_3^1$	AACATACCCCTAATC
$l_1^0$	TCTAATATAATTACT	$l_1^1$	ATTCACTTCTTTAAT
$u_4^0$	AAAACCTACCCTCCT	$u_4^1$	TTTCAATAACACCTC
$l_2^0$	TAATTCCATAACCTA	$l_2^1$	CTTACAATCTTACCT
$u_5^0$	CCAATTCCAATAATC	$u_5^1$	AAATCTATCTAATTC
$l_3^0$	ACCATCTCCAATTCC	$l_3^1$	TAATCCTAATACTAA
$u_6^0$	TTTCCAACCTCCTTCA	$u_6^1$	AATTACCTATTAATC
$l_4^0$	CTACTACAAATCCAC	$l_4^1$	ATTCTATCTTAAACC
$l_5^0$	TATCTACTAAACCAA	$l_5^1$	ATTAATCCTTCAAAC
$u_7^0$	TAATACCTAATTACC	$u_7^1$	AACCCTTACCTACCT
$l_6^0$	TCCACCTTTAATTCC	$l_6^1$	ATTCTAATCCAATT
$u_8^0$	CCAATTTCAACCTAA	$u_8^1$	AATACCTATTACCTT
$l_7^0$	TCCCACAACCTTTTC	$l_7^1$	ATTCTCTCTATAAAT
$u_9^0$	CCTCCTTAATCTACC	$u_9^1$	AACCATACTCTTCAA
$l_8^0$	AATTCCAATTCAATCC	$l_8^1$	TTCCACTTCATTCAA

#### 4.2 算法求解过程

当编码完全产生后,即可对算法进行实现. 算法 3 中各步求解过程如表 2 所示. 其中  $z_i$  为加法运算中第  $i - 1$  次运算产生的进位.

表 2 DNA 计算机新算法各步骤的求解过程

步骤	链	
	$T_{01}$	$T_{02}$
1	$\{x_1^0, x_2^0; x_1^0, x_2^0; x_1^1, x_2^0; x_1^1, x_2^1\}$	$\{x_1^0, x_2^0; x_1^0, x_2^1; x_1^1, x_2^0; x_1^1, x_2^1\}$
2	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1\}$	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1\}$
3	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, u_1^1, u_2^0, u_3^1; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, u_1^1, u_2^0, u_3^1; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, u_1^1, u_2^0, u_3^1; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, u_1^1, u_2^0, u_3^1\}$	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1\}$
4	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^1, l_6^0, u_8^0, l_7^0, u_9^1, l_8^0; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^1, l_6^0, u_8^0, l_7^0, u_9^1, l_8^0; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^1, l_6^0, u_8^0, l_7^0, u_9^1, l_8^0; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^1, l_6^0, u_8^0, l_7^0, u_9^1, l_8^0\}$	$\{x_1^0, x_2^0, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, y_1^0, y_2^0, y_3^0, z_1^0, y_4^0, z_2^0, y_5^0, z_3^0, y_6^0, z_4^0, z_5^0, y_7^0, z_6^0, y_8^0, z_7^0, y_9^0, z_8^0; x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, y_1^0, y_2^0, y_3^0, z_1^0, y_4^0, z_2^0, y_5^0, z_3^0, y_6^0, z_4^0, z_5^0, y_7^0, z_6^0, y_8^0, z_7^0, y_9^0, z_8^0; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, y_1^0, y_2^0, y_3^0, z_1^0, y_4^1, z_2^0, y_5^1, z_3^0, y_6^1, z_4^1, z_5^1, y_7^1, z_6^1, y_8^1, z_7^1, y_9^1, z_8^1; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, y_1^0, y_2^0, y_3^0, z_1^0, y_4^1, z_2^0, y_5^1, z_3^0, y_6^1, z_4^1, z_5^1, y_7^1, z_6^1, y_8^1, z_7^1, y_9^1, z_8^1\}$
5	来自原 $T_{01}$ : $\{x_1^0, x_2^1, s_{1,1}^0, s_{1,2}^0, s_{1,3}^0, s_{2,1}^0, s_{2,2}^0, s_{2,3}^0, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^1, l_6^0, u_8^0, l_7^0, u_9^1, l_8^0+2^0\text{ bp}+2^1\text{ bpDNA 片段}; x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, u_1^1, u_2^0, u_3^1, l_1^0, u_4^1, l_2^0, u_5^0, l_3^0, u_6^1, l_4^0, l_5^0, u_7^0, l_6^0, u_8^0, l_7^0, u_9^0, l_8^0+2^2\text{ bpDNA 片段}\}$ 来自原 $T_{02}$ : $\{x_1^1, x_2^0, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, y_1^0, y_2^0, y_3^0, z_1^0, y_4^1, z_2^0, y_5^1, z_3^0, y_6^0, z_4^0, z_5^0, y_7^1, z_6^0, y_8^1, z_7^1, y_9^1, z_8^0+2^0\text{ bp}+2^1\text{ bpDNA 片段}; x_1^1, x_2^1, s_{1,1}^1, s_{1,2}^1, s_{1,3}^1, s_{2,1}^1, s_{2,2}^1, s_{2,3}^1, y_1^0, y_2^0, y_3^0, z_1^0, y_4^1, z_2^0, y_5^1, z_3^0, y_6^0, z_4^0, z_5^0, y_7^0, z_6^0, y_8^0, z_7^0, y_9^0, z_8^0+2^2\text{ bpDNA 片段}\}$	

由表 2 可知,第 5 步经过凝胶电泳之后,将  $T_{01}$  中的分子按照链长大小进行分离,并通过激光共焦距显微镜观察链长相等的 DNA 链,发现存在链长相等且有 2 种颜色的 DNA 链,它们是差(和)为 2 和 3 的两种链,通过试管  $P_1$ (第 5 步之前  $T_{01}$  的复制)和  $P_2$ (第 5 步之前  $T_{02}$  的复制)提取出差(和)的二进制位为 010 和 110 的 DNA 链,得到试管  $P_1$  表示  $W_1$  中的子集{2}与试管  $P_2$  表示  $W_2$  中的子集{3}的 DNA 链;试管  $P_1$  表示  $W_1$  中的子集{1}与试管  $P_2$  表示  $W_2$  中的子集{4}的 DNA 链.即集合{2,3}和集合{1,4}都为子集和问题的解.

5 结 论

自 DNA 新型计算模型被引入以来,DNA 生物计算机模型一直在不断发展和完善中.理论上,现有计算模型可解决多数计算上的难解问题,且已设计出求解这些问题相应的 DNA 计算机算法.但基于现有模型的算法多使用穷举方法,使得现有 DNA 计算机算法可扩展性过差,直接导致了 DNA 分子生物计算中 DNA 分子链的指数爆炸问题.本文对这一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 完全的子集和问题的

并行求解需求,设计了一种子集和问题的 DNA 计算机模型,该模型不仅具有和其它 DNA 计算机模型同样的生化操作可行性,且能实现子集和二表搜索阶段的内在并行性;在此基础上,利用分治法这一传统并行算法设计技术,提出了一种求解子集和问题的亚指数 DNA 链数和多项式操作时间的 DNA 计算机算法,因此,基于现有生化技术,本算法将子集和问题的求解规模从此前理论上的 60 维扩大到了 120 维.

应该指出,本文模型的研究完全基于求解子集和问题亚指数 DNA 链数的算法设计需求,也仅给出了子集和问题的生物计算实验,但该模型是否还可应用于其它著名 NP 完全问题亚指数链数的 DNA 计算机算法的设计中?如能否设计出基于该模型的  $O(1.26^n)$  链数的团问题 DNA 计算机算法?尽管目前尚不清楚 DNA 生物计算的确切前景,但注意到近年来国内外在 DNA 计算机实际实现上的多种进展,对这些问题进一步的深入研究无疑具有相当意义.

参 考 文 献

[1] Adleman L. Molecular computation of solutions to combinatorial problems. Science, 1994, 266(5187): 1021-1024

- [2] Xu Jin, Huang Bu-Yi. DNA computer principle, advances and difficulties (II): Setting up the database of DNA computer by the synthesis of DNA molecules. Chinese Journal of Computers, 2005, 28(10): 1583-1591(in Chinese)  
(许进, 黄布毅. DNA 计算机: 原理、进展及难点(II) 计算机“数据库”的形成—DNA 分子的合成问题. 计算机学报, 2005, 28(10): 1583-1591)
- [3] Braich R S, Chelyapov N, Johnson C. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296(19): 499-502
- [4] Chang W L, Guo M, Michael H. Fast parallel molecular algorithms for DNA-based computation. IEEE Transactions on Nanobioscience, 2005, 4(2): 133-163
- [5] Li Y, Fang C, Ouyang Q. Genetic algorithm in DNA computing: A solution to the maximal clique problem. Chinese Science Bulletin, 2004, 49(9): 967-971
- [6] Michael H. Fast parallel molecular solutions for DNA-based supercomputing: The subset-product problem. BioSystems, 2005, 80(3): 233-250
- [7] Sinden R R. DNA Structure and Function. London: Academic Press, 1994
- [8] Horowitz E, Sahni S. Computing partitions with applications to the knapsack problem. Journal of ACM, 1974, 21(2): 277-292
- [9] Schroepel R, Shamir A. A  $T=O(2^{n/2})$ ,  $S=O(2^{n/4})$  algorithm for certain NP-complete problems. SIAM Journal on Computing, 1981, 10(3): 456-464
- [10] Bach E, Condon A, Glaser E, Tanguay C. DNA models and algorithms for NP-complete problems. Journal of Computer and System Sciences, 1998, 57(2): 172-186
- [11] Fu B. Volume bounded molecular computation [Ph. D. dissertation]. Department of Computer Science, Yale University, New Haven, Connecticut, USA, 1997
- [12] Natasa J. Trends in computing with DNA. Journal of Computer Science and Technology, 2004, 19(1): 98-114
- [13] Chang W L, Guo M. Molecular solutions for the subset-sum problem on DNA-based supercomputing. BioSystems, 2004, 73(2): 117-130
- [14] Yin Z X, Zhang F Y, Xu J. The general form of 0-1 programming problem based on DNA computing. Biosystems, 2003, 70(1): 73-78
- [15] Lai H C S, Lee J Y, Harn L, Su Y K. Linearly shift knapsack public-key cryptosystem. IEEE Journal Selected Areas Communication, 1989, 7(4): 534-539
- [16] Li K L, Li R F, Li Q H. Optimal parallel algorithm for the knapsack problem without memory conflicts. Journal of Computer Science and Technology, 2004, 19(6): 760-768
- [17] Li Ken-Li, Yao Feng-Juan, Li Ren-Fa, Xu Jin. Improved molecular solutions for the knapsack problem on DNA-based supercomputing. Journal of Computer Research and Development, 2007, 44(6): 1063-1070(in Chinese)  
(李肯立, 姚凤娟, 李仁发, 许进. 基于分治的背包问题 DNA 计算机算法. 计算机研究与发展, 2007, 44(6): 1063-1070)
- [18] Xu J, Li S P, Dong Y F. Sticker DNA computer model-Part I: Theory. Chinese Science Bulletin, 2004, 49(3): 205-212



**LI Ken-Li**, born in 1971, Ph. D. and professor. His main research interests include parallel computing, and molecular computing.

**YAO Feng-Juan**, born in 1981, M. S. candidate. Her

research interests focus on DNA computing.

**XU Jin**, born in 1959, professor, Ph. D. supervisor. His research interests include DNA computing and DNA computer, neural networks, genetic algorithms, graph theory.

**LI Ren-Fa**, born in 1957, professor and Ph. D. supervisor. His main research interests include wireless network, mobile computing, embedded computing.

## Background

This research is supported by the key Project of National Natural Science Foundation of China under grant No. 60533010, the Projects of National Natural Science Foundation of China under grants (60603053, 60403002): Research on a scalable DNA computer model, the Theory, Model and Method of DNA Computer etc. The projects mainly focus on DNA computer models for processing graphical messages, including encoding DNA sequences, synthesi-

zing DNA molecules, setting up the model, detecting solutions, etc. The research group has been working on many aspects of DNA computing since 1996 and have published a monograph and more than 120 papers on DNA computing and DNA computer. This paper uses the basic strategy for devising parallel algorithm—Divide and conquer to design a proposed DNA model of subset-sum problem for avoidance of the limitation of enumeration in subset-sum problem.