计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
45(10): 1782-1788, 2008

精确覆盖问题的 $O(1.414^n)$ 链数 DNA 计算机算法

李肯立[1] 刘 杰[1] 杨 磊[1] 刘文斌[2]
[1](湖南大学计算机与通信学院 长沙 410082)
[2](温州大学计算机科学与工程学院 浙江温州 325027)
(lkl510@263.net)

An $O(1.414^n)$ Volume Molecular Solutions for the Exact Cover Problem on DNA-Based Supercomputing

Li Kenli[1], Liu Jie[1], Yang Lei[1], and Liu Wenbin[2]
[1](*College of Computer and Communication, Hunan University, Changsha* 410082)
[2](*College of Computer Science and Engineering, Wenzhou University, Wenzhou, Zhejiang* 325027)

Abstract The scalability problem in DNA computer has been one of the important research areas in DNA computing. According to the requirement of the DNA parallel computing for exact cover problem, a DNA model for good scalability is proposed, which is based on the biological operations in the Adleman-Lipton model and the solution space of stickers in the sticker-based model by simultaneously taking the method of fluorescence labeling and the technique of gel electrophoresis into the model. Based on this model, a DNA-based algorithm for the exact cover problem, by making use of the strategem of divide-and-conquer, is also proposed which consists of three sub-algorithms: *Init*(), *IllegalRemove*(), and *ParallelSeacher*(). Compared with by far the best molecular algorithm for the exact cover problem with n variables and m sets in which $O(2^n)$ DNA strands are used, this algorithm can solve the same problem only using $O(1.414^n)$ DNA strands on the condition of not varying the time complexity. Therefore, the cardinal number of the exact cover problem that can be theoretically resolved in a test tube may be enlarged from 60 to 120, and thus the proposed algorithm is an improved result over the past researches.

Key words DNA computer; NP-complete problem; exact cover problem; divide and conquer; DNA super-computing

摘 要 DNA 计算机的可扩展性问题是近年来生物计算领域的重要研究重点之一.根据精确覆盖问题 DNA 计算求解过程中的并行计算需求,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,提出了一种求解精确覆盖问题的 DNA 计算模型和基于分治方法的 DNA 计算机算法.算法由初始解空间生成算法 *Init*()、冗余解删除算法 *IllegalRemove*()和并行搜索器 *ParallelSeacher*()共 3 个子算法组成.与同类算法的性能比较分析表明:本算法在保持多项式生物操作复杂性的条件下,将求解 n 维精确覆盖问题的 DNA 链数从 $O(2^n)$减少至 $O(1.414^n)$,从而将 DNA 计算机在试管内可求解的精确覆盖问题集合的基数从 60 提高到 120,改进了相关文献的研究结果.

关键词 DNA 计算机; NP 完全问题; 精确覆盖问题; 分治法; DNA 超级计算

中图法分类号 TP301.6

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

收稿日期:2007-09-03;**修回日期**:2008-03-28
基金项目:国家自然科学基金项目(60603053,90715029);浙江省自然科学基金项目(106654);高等学校博士后科学基金项目(20060400845)

Hamilton 路径问题[1]. 1995 年,Lipton 成功求解了第 1 个 NP 完全问题——可满足性(SAT)问题[2]. 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,其相关研究形成了理论计算机科学的一个新的研究热点,有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[3-9].

然而,目前大多数 NP 完全问题都是基于穷举的方法,使得初始解空间随着问题规模的增大而呈指数性增大. 在传统计算机领域里,我们发现多数 NP 完全问题都已有比蛮力搜索方法更好的亚指数时间算法[10-12]. 如能设计出和这些亚指数时间算法相应的 DNA 计算机算法,将大大提高 DNA 计算机算法的可扩展性. 2004 年,李源等人[8]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;文献[13]则提出了一种基于传统分治策略的背包问题 DNA 计算机算法. 上述两种算法均可大大减少直接穷举方法中试管内 DNA 的分子数目,使得算法的可扩展性得到一定改善,但都存在某些不足:前者的成功仅限于问题规模不太大的实例;后者的生物操作复杂性为伪多项式.

本文对精确覆盖问题 DNA 计算机算法的扩展性进行了较深入的探索. 精确覆盖问题已被证明是 NP 完全问题,描述如下:给定一个集合 $S=(s_1,s_2,\cdots,s_q)$和一个表示 S 子集的集合 $C=(C_1,C_2,\cdots,C_n)$,要求集合 C 的一个子集 C^1,使得集合 C^1 中所有子集互不相交且并集为 S. 2004 年,Chang 等人利用 Adleman-Lipton 模型的操作与粘贴模型的解空间相结合,设计了 $O(2^n)$链数求解 3-集合精确覆盖问题的 DNA 计算机算法[14]. 该算法证明了精确覆盖问题可通过 DNA 计算完成,但由于该算法同样基于穷举方法,以现有生化技术,在试管内仅能求解基数为 60 的精确覆盖问题.

为提高基于 DNA 计算的精确覆盖问题解搜索阶段的并行度,本文将 Adleman-Lipton 模型的操作与粘贴模型的解空间将结合,提出了一种求解精确覆盖问题的 DNA 计算机模型和基于此模型的算法. 新算法由初始解空间生成算法、冗余解删除算法和并行搜索器共 3 个子算法组成. 和文献[13-14]的算法相比,新算法在不增加操作复杂性的条件下将求解该问题所需的 DNA 链数从纯指数的 $O(2^n)$减少至亚指数的 $O(1.414^n)$.

1 精确覆盖问题 DNA 计算模型

Adleman-Lipton 模型的生物操作与粘贴模型(sticker model)的解空间相结合的模型是一种通用的 DNA 计算模型[7,13-16],它兼具 Adleman-Lipton 模型和粘贴模型(sticker model)的优点,可随机存储和具有较低的杂交误解率. 基于此计算模型,本文对精确覆盖问题的求解所采用的基本生化操作另外增加两种(可行性分析具体参考文献[16]). 生物操作描述如下[16]:

① 抽取(extract). 给定试管 P 和一个短的 DNA 链 S. 抽取操作有:$+(P,S)$和$-(P,S)$. $+(P,S)$表示 P 中所有包含 S 作为子链的 DNA 分子链. $-(P,S)$表示 P 中所有不包含 S 作为子链的 DNA 分子链.

② 合并(merge). 给定试管 P_1 和 P_2,操作 $\cup(P_1,P_2)=P_1\cup P_2$ 表示将试管 P_1 和 P_2 合并到一个试管中而不改变 P_1 和 P_2 中的任何链.

③ 检测(detect). 给定试管 P,如果 P 中至少包含一个 DNA 链,则返回“yes”,否则,返回“no”.

④ 复制(amplify). 给定试管 P,操作 $Amplify(P,P_1,P_2)$将产生 P 的两份复制 P_1 和 P_2,且之后试管 P 为空.

⑤ 退火(anneal). 给定试管 P,操作 $Anneal(P)$将试管 P 中的所有 DNA 单链变成双链.

⑥ 添加(append). 给定试管 P 和一个短的 DNA 链 Z,该操作将链 Z 添加到 P 中每个链的末尾.

⑦ 读取(read). 给定试管 P,该操作表示读取 P 中每个 DNA 分子链的所有 0/1 信息.

⑧ 荧光标记(fluorescence labeling). 给定试管 P,该操作将具有某种特征的 DNA 分子作标记.

⑨ 凝胶电泳(gel electrophoresis). 给定试管 P,该操作将试管中的 DNA 链按照分子大小进行分离.

2 精确覆盖问题 DNA 计算机算法

2.1 基于分治的 DNA 计算机算法思想

1974 年,Horowitz 和 Sahni[10]将分治方法引入背包问题的算法设计中,提出了使用分治的著名的二表算法,其主要步骤如下[10]:

① 将集合 W 均分成两部分,生成两部分的所

有子集和放入表 A,B 中;

② 对表 A,B 中的所有元素排序;

③ 对表 A,B 进行搜索以得到问题的解.

该算法的主要贡献是将求解子集和问题的时间复杂性从 $O(2^n)$降低到 $O(2^{n/2})\approx 1.414^n$,参见文献[10].那么,能否将分治法引入到对精确覆盖问题的求解当中并提出相应的 DNA 算法,以降低求解此问题 DNA 现有计算机算法纯指数增长的 DNA 链数? 对背包问题的初步尝试表明了这一途径是可行的[13],但该算法仍存在一定局限,因为该算法在将求解背包问题 DNA 计算机算法的链数降低至 $O(2^{n/2})$的同时,算法的生化实验操作次数却从多项式变成了伪多项式,而伪多项式的生物操作复杂性将大大降低 DNA 算法的实验可行性. 为了克服这一局限,本文对精确覆盖问题的 DNA 计算机算法进行了更深入的探索与研究,通过对 Adleman-Lipton 模型的生物操作进行完善,并受 Horowitz 和 Sahni 二表算法设计思想的启示[10],在 DNA 分子算法设计中也引入分治策略,但不直接用 DNA 分子操作实现基于分治的算法,而根据 DNA 分子操作的特性,以一种巧妙的方式达到既减少算法中的 DNA 链数,又不致使 DNA 分子操作复杂性从多项式增加到伪多项式,从而克服前述方法中生化操作复杂性过高的缺点.

本算法的基本步骤为:利用分治策略,将集合 C 中所有子集分量 $C_1, C_2, \cdots, C_n$ 均分成两部分,$W_1=(C_1, C_2, \cdots, C_{n/2})$和 $W_2=(C_{n/2+1}, C_{n/2+2}, \cdots, C_n)$. 分别求出 $W_1(W_2)$的全部 $2^{n/2}$ 个子集,记作 $SW_1(SW_2)$. 再分别删除其中表示冗余解的子集,即 $SW_1(SW_2)$中的子集中子集分量有交集的情况,记作 $SW_1'(SW_2')$. 然后根据精确覆盖问题的定义,采用与文献[14]不同的并行搜索算法,并行地搜索分别来自于 SW_1' 与 SW_2' 中的一对子集,使得所包含的集合 S 中各元素 $s_1, s_2, \cdots, s_q$ 恰好互为补集,此时,这对子集的并集即为所求的精确覆盖问题的解. 注意到 DNA 操作的生物特性,本算法用 DNA 分子操作的基本实现过程为:

算法 1. 精确覆盖问题的 DNA 算法框架(DNA-based algorithm frame for exact cover problem).

1) 分别在 T_{01} 和 T_{02} 中用 DNA 链表示 $W_1=(C_1, C_2, \cdots, C_{n/2})$和 $W_2=(C_{n/2+1}, C_{n/2+2}, \cdots, C_n)$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,而在 T_{02} 中每个不同的 DNA 链的前 $n/2$ 个不同的子段分别连接上的同一种其他颜色的荧光素.

2) 分别在 T_{01} 和 T_{02} 中通过在 DNA 链末尾添加 DNA 子链的方式表示 W_1 和 W_2 的所有子集的对应元素,并且在此过程中去除表示冗余解的 DNA 链(子集 W_1/W_2 中对应子集分量有交集的情况).

3) 解搜索过程分别将 T_{01} 中 DNA 链上对应集合 S 中所有元素的二进制的取反信息和 T_{02} 中 DNA 链对应 S 中所有元素的二进制信息转换成相应的链长,使用荧光标记和凝胶电泳技术,利用并行搜索器进行搜索,如果能够找到等长且有两种颜色的一对 DNA 链,则问题有解,此时等长的 DNA 链上表示的子集 W_1 和子集 W_2 的并集即为精确覆盖问题的解;否则问题无解.

在算法 1 中,为使解搜索中能够利用 DNA 计算的内在并行性,步骤 1)在初始生成 W_1 和 W_2 所有子集的过程中(可利用文献[15]的 *Init*()子程序,囿于篇幅,本文不再赘叙),采用荧光标记方法,将试管 T_{01} 和 T_{02} 中每个不同 DNA 链的前 $n/2$ 个不同子段分别连接上红色荧光素和绿色荧光素,其目的是使得之后的并行搜索器算法能以多项式的操作次数实现;步骤 2)去除冗余解采用 *IllegalRemove*()算法;步骤 3)中并行搜索器由算法 *ParallelSearcher*()实现. 以下详叙这两个 DNA 计算机子算法的设计.

2.2 冗余解的删除算法

算法 1 第 1 步产生了链数均为 $2^{n/2}$ 的两个试管 T_{01} 和 T_{02},分别代表集合 C 的前半部分和后半部分的解空间. 根据算法 1 的第 2)步,对于试管 T_{01} 和 T_{02} 中的 DNA 链,应通过基本生物操作来删除其中表示冗余解的链. 令 C_k 表示集合 C 中的第 k 个子集. 若 C^1 中包含子集 C_k,则用 C_k^1 标记,否则用 C_k^0 标记. 同样,令 s_j 表示集合 S 中的第 j 个元素. 若 C^1 中包含元素 s_j,则用 s_j^1 标记,否则用 s_j^0 标记.

算法 2. 冗余解删除算法(parallel deletion algorithm for illegal solutions).

Procedure *IllegalRemove* (T_{01}, $n/2$, q)

1) For $k=1$ to $n/2$

1.1 $T_{ON}=+(T_{01}, C_k^1)$ and $T_{OFF}=-(T_{01}, C_k^1)$

1.2 For $i=1$ to $|C_k|$

Assume that the ith element in C_k is the j th element s_j in S.

1.3 $T_{BAD}=+(T_{ON}, s_j^1)$ and $T_{ON}=-(T_{ON}, s_j^1)$

1.4 *Discard*(T_{BAD})

1.5 *Append*(T_{ON}, s_j^1)

EndFor

1.6 $T_{01}=\cup(T_{ON}, T_{OFF})$

EndFor

2) For $i=1$ to q

2.1 $T_{ON}=+(T_{01}, s_j^1)$ and $T_{OFF}=-(T_{01}, s_j^1)$

2.2 $Append(T_{OFF}, s_j^0)$

2.3 $T_{01}=\cup(T_{ON}, T_{OFF})$

EndFor

EndProcedure

引理 1. 算法 $IllegalRemove(T_{01}/T_{02}, n/2, q)$ 将并行删除子集 W_1 和 W_2 中那些对应于集合 C 中各子集分量有交集的 DNA 链.

证明. 在执行 $Init(T_{01}, n/2)$ 算法之后,试管 T_{01} 中含有 $2^{n/2}$ 个 DNA 链,表示 W_1 的所有 $2^{n/2}$ 个子集.算法 $IllegalRemove(T_{01}, n/2, q)$ 的第 1)步循环删除表示冗余解的 DNA 链(子集 W_1 中子集分量有交集的情况),并在此过程中不断在 DNA 链末尾添加其所对应的 S 中的元素. 其中,步骤 1.1 采用抽取操作将 DNA 链按照子集 C_k 为'1'与'0'的情况,分别分离到试管 T_{ON} 和 T_{OFF} 中. 步骤 1.2 循环遍历子集 C_k 中的所有元素.步骤 1.3 采用抽取操作将 DNA 链按照元素 s_i 为'1'与'0'的情况,分别分离到试管 T_{BAD} 和 T_{ON} 中,因此试管 T_{BAD} 表示已有另一个子集 C_k 也包含元素 s_i.步骤 1.4 抛弃试管 T_{BAD} 中的 DNA 链,因为这些 DNA 链表示的是冗余解.步骤 1.5 在试管 T_{ON} 的 DNA 链末尾添加表示元素 s_i 为'1'的 DNA 序列.步骤 1.6 采用合并操作将试管 T_{ON} 和 T_{OFF} 再重新合并到试管 T_{01} 中. 循环结束后,试管 T_{01} 中只保留了表示合法解的 DNA 链,并且其对应包含的所有 S 中的元素都将被添加上去.

第 2)步循环是为表示合法解的 DNA 链补充添加其不包含的 S 中的元素.步骤 2.1 采用抽取操作将 DNA 链按照元素 s_i 为'1'与'0'的情况,分别分离到试管 T_{ON} 和 T_{OFF} 中.步骤 2.2 采用 $Append()$ 操作在试管 T_{OFF} 中的所有 DNA 链的末尾添加表示元素 s_i 为'0'的 DNA 序列.步骤 2.3 采用合并操作将试管 T_{ON} 和 T_{OFF} 再重新合并到试管 T_{01} 中. 循环结束后,试管 T_{01} 中的 DNA 链的构造如图 1 所示:

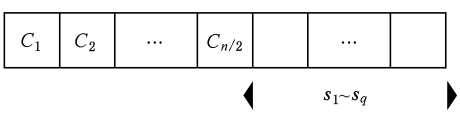


Fig. 1 Structure of DNA strands in tube T_{01} after algorithm 2 is performed.

图 1 执行算法 2 后试管 T_{01} 中的 DNA 链结构

算法 $IllegalRemove(T_{01}, n/2, q)$ 中,最坏情况下须使用 $n/2\times(1+q)+q$ 次抽取、$(n/2+1)\times q$ 次添加、$n/2+q$ 次合并操作来完成冗余解的删除和对应元素的添加.算法所使用的试管数为 4,DNA 链的最大链长为 $n/2+q$.

2.3 并行搜索器

根据算法 1 的第 3)步,接下来需要在试管 T_{01} 和 T_{02} 中寻找对应 S 中的元素 $s_1,\cdots,s_q$ 所表示的二进制值恰好互补的 DNA 链,如果能够找到这样的 DNA 链,则它们所分别对应的子集的并集即为精确覆盖问题的解.下述算法 3 将完成这一并行搜索工作.

算法 3. 解并行搜索器(parallel searcher for feasible solutions).

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

1) For $j=1$ to q

1.1 $T_1=+(T_{01}, s_j^0)$ and $T_2=-(T_{01}, s_j^0)$

1.2 $T_3=+(T_{02}, s_j^1)$ and $T_4=-(T_{02}, s_j^1)$

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

1.4 Append a double-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) $Amplify(T_{01}, T_{01}, P_1)$

3) $Amplify(T_{02}, T_{02}, P_2)$

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

5) $Anneal(T_{01})$

6) 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,通过激光共焦距显微镜观察链长相等的 DNA 链,将具有两种不同颜色的 DNA 链分离出来,得到表示 W_1 某子集的补集信息与 W_2 某子集的并集信息的一对 DNA 链,其链长记为 l,然后从试管 P_1 和 P_2 中,分别分离出与链长为 l 的一对 DNA 单链,其分别对应的子集 W_1 和子集 W_2 的并集即为精确覆盖问题的解.

EndProcedure

引理 2. 算法 $ParallelSearcher(T_{01}, T_{02})$ 可搜索出 T_{01} 中各 DNA 链表示的 $s_1,\cdots,s_q$ 的值与 T_{02} 中各 DNA 链表示的 $s_1,\cdots,s_q$ 的值互补的链.

证明. 算法 $ParallelSearcher(T_{01}, T_{02})$ 中第 1)步将循环 q 次,其目的是将试管 T_{01} 和 T_{02} 中 s_1 至 s_q 的二进制信息转换成相应的链长信息.步骤 1.1 使用抽取操作,将 T_{01} 中的 DNA 链按照元素 s_j 为

'1'与'0'的情况,分别分离到试管 T_1 和 T_2 中;步骤 1.2 使用抽取操作,将 T_{02} 中的 DNA 链按照元素 s_j 为'1'与'0'的情况,分别分离到试管 T_3 和 T_4 中. 步骤 1.3 和步骤 1.4 分别使用添加操作,将一段链长为 2^{j-1} 的 DNA 双链分别添加到试管 T_1 和 T_3 的每条链的末尾. 步骤 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} 和 T_{02} 中 s_1 至 s_q 的二进制信息都转换成了相应的链长信息,即每条链上的 s_1 至 s_q 的二进制信息都有一个唯一的链长值与其对应.

算法的第 2)步和第 3)步使用复制操作分别将试管 T_{01} 和 T_{02} 复制于试管 P_1 和 P_2 中. 算法的第 4)步将试管 T_{01} 与 T_{02} 合并到 T_{01}. 第 5)步使用退火操作将试管 T_{01} 中的单链变成双链. 第 6)步使用凝胶电泳技术将试管 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是否存在颜色不同且链长相等的 DNA 链,若存在,则它们在 W_1 和 W_2 中分别对应的子集的并集即为精确覆盖问题的解.

算法 $ParallelSearcher(T_{01}, T_{02})$ 需使用 $2q$ 次抽取、$2q$ 次添加、$2q+1$ 次合并、2 次复制和 1 次退火操作完成并行搜索. 由于执行算法 2 后,其链长为 $O(n/2+q)$,因此,执行 $ParallelSearcher(T_{01}, T_{02})$ 后,最大链长将变为 $O(n/2+q+2^q)$,使用的试管数为 8.

2.4 精确覆盖问题 $O(1.414^n)$ 链数 DNA 算法

将实现步骤 1)~3)的前叙各子算法组成一个整体,得到基于 DNA 超级计算的精确覆盖问题求解算法.

算法 4. $O(1.414^n)$ 链数求解精确覆盖问题的 DNA 算法(DNA algorithm for exacting cover problem)

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

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

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

定理 1. 算法 4 可通过 DNA 生物分子超级并行的方式求解精确覆盖问题.

证明. 算法第 1)步执行 $Init(T_{01}, n/2)$ 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)步使用 $IllegalRemove(T_{01}, n/2, q)$ 和 $IllegalRemove(T_{02}, n/2, q)$ 将集合 W_1 和 W_2 的子集中表示冗余解的 DNA 链进行了删除. 第 3)步执行 $ParallelSearcher(T_{01}, T_{02})$ 并行搜索 T_{01} 中各 DNA 链表示的 S 中元素 $s_1, \cdots, s_q$ 的二进制值与 T_{02} 中各 DNA 链表示的 $s_1, \cdots, s_q$ 的二进制值恰好相反的链,若搜索成功,则所找到的成对的 DNA 链上所对应的 W_1 和 W_2 子集的并集即为精确覆盖问题的解,否则,该精确覆盖问题实例无解.

2.5 性能分析与比较

衡量 DNA 计算机算法的性能的标准通常包括 DNA 生物分子操作的时间复杂性和空间复杂性[7-8,13-16],时间复杂性主要表现为算法的生化实验操作次数,由于分子生物学实验技术的限制,空间复杂性首先包括算法中试管中的 DNA 分子链数以及所使用的实验试管总数等. 本文算法的复杂性为:

定理 2 在精确覆盖问题中,若集合 S 和 C 的基数分别为 q 和 n,则算法 4 求解该问题实例所用到的生物操作数为 $O(qn)$,测试试管数为常数,DNA 链数为 $O(1.414^n)$.

证明. 算法 4 共由 5 个子算法组成. 由引理 1 和 2 知,总的生物操作数为 $O(qn)$,测试试管总数为 12,即 $O(1)$. 利用分治策略,算法在第 1)步中将包含 n 个子集的集合 C 平均分成两个各包含 $n/2$ 个子集的集合 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,此后的算法执行过程中没有生成新的 DNA 链,因此,算法总的 DNA 链数为 $O(2^{n/2})$ 即 $O(\sqrt{2}^n) \approx 1.414^n$.

文献[14]中求解 3-集合精确覆盖(集合 S 和 C 的基数分别为 $3q$ 和 n,且集合 C 中的各元素(集合)的基数都为 3)问题的 DNA 计算机算法所需的生物操作数为 $O(3n+3q)$,DNA 链数为 $O(2^n)$. 文献[14]基于分治的背包问题 DNA 算法中,虽然将 DNA 链数从文献[13]的 $O(2^n)$ 降低到 $O(2^{n/2})$,但算法的生物操作复杂度却从多项式变成了伪多项式. 文献[17]通过改进 DNA 编码方案可使 0/1 背包问题的 DNA 计算机算法的时间复杂度为 $O(n^2)$,但 DNA 链数仍为 $O(2^n)$. 本文所提出的精确覆盖 DNA 计算模型求解 n 维精确覆盖的 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 $O(qn)$,提高了通过实际生化实验求解大规模精确覆盖问题的可能性.

值得指出的是,和电子计算机的计算速度相比,目前生物分子计算机中实际生化反应的单次生物操作需要更长的时间,而本文算法的生化操作次数 $O(qn)$ 多于文献[15]的 $O(3n+3q)$. 但由于现有生

化实验技术允许的单个试管中的 DNA 分子浓度(即链数)仍然有限[1-2,18]. 因此,和上述相关算法相比,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仍是明显的.

3 DNA 编码及算法实现

以 $S=\{1,2,3\}$,$C=\{\{1,3\},\{3\},\{1\},\{2\}\}$作为精确覆盖问题实例,以下给出使用本算法对此实例的模拟求解过程.

首先,采用 Braich 等人求 20 个变量 SAT 问题中使用的 DNA 计算模型[5],对每一变量设计两个长度均为 15 的碱基"值序列". 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. 其中表 1 是算法中所有 7 个变量的 DNA 序列. C_i 为集合 C 中第 i 个子集分量的标号,若 C^1 中包含子集 C_k,则用 C_k^1 标记,否则用 C_k^0 标记. 同样,s_j 表示集合 S 中的第 j 个元素. 若 C^1 中包含元素 s_j,则用 s_j^1 标记,否则用 s_j^0 标记. 算法 4 中各主要步骤的求解过程如表 2 所示.

由表 2 可知,在经过第 5)步的凝胶电泳之后,T_{01}中的分子已按照链长大小进行分离,此时通过激光共焦距显微镜观察链长相等的 DNA 链,发现存在两对 DNA 链,每对 DNA 链都由两种颜色组成,且其中一对链长为 78bps,另一对链长为 77bps. 从试管 P_1 和 P_2(算法 3 第 3)步对 T_{01}和 T_{02}的复制)中分别提取出链长为 78bps 和 77bps 的两对 DNA 单链,再从试管分别提取出链长为 78bps 和 77bps 的两条 DNA 单链,然后读出具有相同链长的两对 DNA 链上所对应的 $C_1C_2C_3C_4$ 值分别为 0111 和 1001,因此,此精确覆盖问题实例的解为:{{3},{1},{2}}与{{1,3},{2}}.

Table 1 DNA Sequences for 7 Variables

表 1 7 个变量的 DNA 序列

bit	5′→3′ DNA Sequence	bit	5′→3′ DNA Sequence
C_1^0	TATCTACTAAACCAA	C_1^1	ATTAATCCTTCAAAC
C_2^0	TAATACCTAATTACC	C_2^1	AACCCTTACCTACCT
C_3^0	TCCACCTTTAATTCC	C_3^1	ATTCCTAATCCAATT
C_4^0	CCAATTTCAACCTAA	C_4^1	AATACCTATTACCTT
s_1^0	TCCCACAACTCTTTC	s_1^1	ATTCCTCCTATAAAT
s_2^0	CCTCCTTAATCTACC	s_2^1	AACCATACTCTTCAA
s_3^0	AATTCCATTCAATCC	s_3^1	TTCCACTTCATTCAA

Table 2 Execution Course of Algorithm 4

表 2 DNA 计算机新算法各步骤的求解过程

Step	Tube	
	T_{01}	T_{02}
1)	$\{C_1^0,C_2^0;C_1^0,C_2^1;$ $C_1^1,C_2^0;C_1^1,C_2^1\}$	$\{C_3^0,C_4^0;C_3^0,C_4^1;$ $C_3^1,C_4^0;C_3^1,C_4^1\}$
2)	$\{C_1^0,C_2^0,s_1^0,s_2^0,s_3^0;$ $C_1^0,C_2^1,s_3^1,s_1^0,s_2^0;$ $C_1^1,C_2^0,s_1^1,s_3^1,s_2^0\}$	$\{C_3^0,C_4^0,s_1^0,s_2^0,s_3^0;$ $C_3^0,C_4^1,s_2^1,s_1^0,s^0;$ $C_3^1,C_4^0,s_1^1,s_2^0,s_3^0;$ $C_3^1,C_4^1,s_1^1,s_2^1,s_3^0\}$
3)	Two DNA strands separated out by gel electrophoresis are: $\{C_1^0,C_2^1,s_3^1,s_1^0,s_2^0;$ $C_3^1,C_4^1,s_1^1,s_2^1,s_3^0\}$ and $\{C_1^1,C_2^0,$ $s_1^1,s_3^1,s_2^0;$ $C_3^0,C_4^1,s_2^1,s_1^0,s_3^0\}$, therefore the two corresponding solutions for the exact cover problem are: {{3},{1},{2}} and {{1,3},{2}}.	

4 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[13-14,16]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 NP 完全的精确覆盖问题解搜索阶段的并行求解需求和现有生化操作的特性,设计了一种求解精确覆盖问题的 DNA 计算机模型和算法,本算法在保持多项式的生化操作时间的条件下,将求解精确覆盖问题所需的 DNA 分子链数从此前的 2^n 减少到了亚指数的 $2^{n/2}$,理论分析和模拟实验表明了本文算法的正确性. 因此,基于现有生化技术,本算法理论上可将精确覆盖问题的求解规模从 60 维扩大到 120 维.

应该指出,本文算法所采用的算法设计策略以及相应的计算模型,虽可以亚指数链数和多项式时间实现精确覆盖问题的求解,但这种方法和模型是否适应包括 SAT 问题在内的其他 NP 完全问题,尽管目前尚不清楚 DNA 分子超级计算的确切前景,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[5-6],对这些方向的进一步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 [J]. Science, 1994, 266(5187): 1021-1024

[2] Lipton R J. DNA solution of hard computational problems [J]. Science, 1995, 268(5210): 542-545

[3] Xu Jin, Tan Gangjun, Fan Yueke, *et al*. DNA computer principle, advances and difficulties(IV): On the models of DNA computer [J]. Chinese Journal of Computers, 2007, 30(6): 881-893 (in Chinese)
(许进,谭刚军,范月科,等. DNA 计算机原理、进展及难点(IV):论 DNA 计算机模型[J]. 计算机学报, 2007, 30(6): 881-893)

[4] Garzon M H, Deaton R J. Biomolecular computing and programming [J]. IEEE Trans on Evolutionary Computation, 1999, 3(3): 236-250

[5] Braich R S, chelyapov N, Johnson C. Solution of a 20-variable 3-SAT problem on a DNA computer [J]. Science, 2002, 296(19): 499-502

[6] Li Wanggen, Ding Yongsheng. Design and implementation of queue data structure in DNA computer [J]. Chinese Journal of Computers, 2007, 30(6): 993-998 (in Chinese)
(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, 30(6): 993-998)

[7] Chang W L, Guo M, Michael H. Fast parallel molecular algorithms for integer factoring [J]. IEEE Trans on Nanobioscience, 2005, 4(2): 133-163

[8] Li Yuan, Fang Chen, Ouyang Qi. Genetic algorithm in DNA computing: A solution to the maximal clique problem [J]. Science Bulletin, 2004, 49(5): 439-443 (in Chinese)
(李源, 方辰, 欧阳颀. 最大团问题的 DNA 计算机进化算法[J]. 科学通报, 2004, 49(5): 439-443)

[9] Benenson Y, Gil B, Ben-Dor U, *et al*. An autonomous molecular computer for logical control of gene expression [J]. Nature, 2004, 429 (6990):423-429

[10] Horowitz E, Sahni S. Computing partitions with applications to the knapsack problem [J]. Journal of ACM, 1974, 21 (2): 277-292

[11] Bach E, Condon A, Glaser E, *et al*. DNA models and algorithms for NP-complete problems [J]. Journal of Computer and Systems Sciences, 1998, 57(2): 172-186

[12] Fu B. Volume bounded molecular computation [D]. New Haven, Connecticut, USA: Department of Computer Science, Yale University, 1997

[13] Li Kenli, Yao Fengjuan, Li Renfa, *et al*. Improved molecular solutions for the knapsack problem on DNA-based supercomputing [J]. Journal of Computer Research and Development, 2007, 44(6): 1063-1070 (in Chinese)
(李肯立, 姚凤娟, 李仁发, 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)

[14] Chang W L, Guo M. Solving the set cover problem and the problem of exact cover by 3-sets in the Adleman-Lipton model [J]. BioSystems, 2003, 72: 263-275

[15] Chang W L, Guo M. Molecular solutions for the subset-sum problem on DNA-based supercomputing [J]. BioSystems, 2004, 73: 117-130

[16] Li Kenli, Yao Fengjuan, Xu Jin, *et al*. An $O(1.414^n)$ volume molecular solutions for the subset-sum problem on DNA-based supercomputing [J]. Chinese Journal of Computers, 2007, 30(11): 1948-1953 (in Chinese)
(李肯立, 姚凤娟, 许进, 等. 子集和问题的 $O(1.414^n)$链数 DNA 计算机算法[J]. 计算机学报, 2007, 30(11): 1948-1953)

[17] Han Aili, Zhu Daming. DNA computing Model for the 0/1 knapsack problem [C] //Proc of the 6th Int Conf on Hybrid Intelligent Systems(HIS'06). Los Alamitos: IEEE Computer Society, 2006

[18] Lu Shengdong. The Experimentation of Molecular Biology [M]. Berjing: Peking Union Medical College Press, 1999 (in Chinese)
(卢圣东. 现代分子生物实验技术[M]. 北京: 中国协和医科大学出版社, 1999)

Li Kenli, born in 1971. Doctor and professor. Senior member of China Computer Federation. His main research interests include parallel computing and molecular computing.

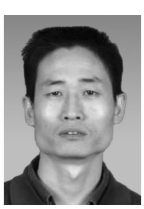
李肯立,1971 年生,博士,教授,中国计算机学会高级会员,主要研究方向为并行计算、生物计算机.

Liu Jie, born in 1985. Master candidate. Her main research interests include DNA computing.

刘 杰,1985 年生,硕士研究生,主要研究方向为 DNA 计算.

Yang Lei, born in 1976. Ph. D. candidate and lecturer. His main research interests include DNA computing and distributed computing.

杨 磊,1976 年生,博士研究生,讲师,主要研究方向为 DNA 计算和分布式计算.



Liu Wenbin, born in 1971. Associate professor. His main research interests include DNA computing and ntelligence computing.

刘文斌,1971 年生,副教授,主要要研究方向为 DNA 计算与智能计算.

Research Background

DNA computing is a new computation paradigm that employs molecule manipulation to solve computational problems especially for NP problems. However, DNA-based direct exhaustion leads to the DNA strands increase exponentially, which becomes the bottleneck factor in the development of DNA computing. How to decrease the number of DNA strands increasing exponentially in these applications is very important in the research on DNA computers. In this paper, the strategy of divide-and-conquer is introduced into the DNA-based supercomputing and a DNA algorithm is proposed. This work is supported in part by the National Natural Science Foundation of China under grant Nos. 60603053, 60503002,60533010, Natural Science Foundation of Zhejiang Province under grant No. Y106654, and the Postdoctoral Science Foundation of China under grant No. 20060400845.