

最大匹配问题的粘贴 DNA 算法

吴雪¹ 宋晨阳² 张楠¹ 朱煜¹ 陈志华¹

(华东理工大学信息科学与工程学院 上海 200237)¹ (上海通用识别技术研究所 上海 201112)²

摘要 最大匹配问题(MMP)是图论中经典的组合优化问题。针对此问题提出了基于 DNA 粘贴计算模型的求解算法,阐述了该算法如何利用 DNA 链构建最大匹配问题的初始编码,说明了应用粘贴计算模型寻求最终解的生物操作过程,同时分析了此 DNA 并行算法的计算复杂度,最后给出了该算法的计算机模拟仿真结果和应用实例,得到了所给问题的最大匹配解,并对算法的可行性进行了验证和总结。

关键词 DNA 计算,最大匹配,粘贴模型

中图分类号 TP301.6 **文献标识码** A

DNA Algorithm for Maximum Matching Problem Based on Sticker Computation Model

WU Xue¹ SONG Chen-yang² ZHANG Nan¹ ZHU Yu¹ CHEN Zhi-hua¹

(School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)¹

(Shanghai General Recognition Technology Institute, Shanghai 201112, China)²

Abstract This paper the DNA solution of the maximum matching problem(MMP) based on sticker computation model was presented, showed how to use DNA strands to construct solution space of molecules for the maximum matching problem and how to apply the biological operations of sticker model to solve the problem from the solution space of molecules, at the same time analysed of the computational complexity for DNA parallel algorithms. Finally, the computer program was given to simulate this algorithm and the solutions of MMP for all examples were also found, and the feasibility of the algorithm was validated and summarized.

Keywords DNA computing, Maximum matching, Sticker model

1 引言

1994 年,南加州大学的 Adleman 博士首次利用分子生物技术,在试管中对 DNA 编码链进行了生物操作实验,解决了一个简单有向图的哈密尔顿路径问题(Hamilton Path Problem, HPP),开创了解决 NP-完全问题的新方法^[1]。随后,一大批科学家在 DNA 分子计算这一领域展开了研究,对组合优化问题中的许多 NP-完全问题提出了各种计算模型。Lipton 在 Adleman 实验的基础上,建立了求解可满足性问题(Satisfaction Problem, SAT)的 DNA 计算模型^[2]。Ouyang 等提出了求解图的最大团问题(Maximal Clique Problem, MCP)的 DNA 计算模型^[3];Head 等提出了最大独立集(Maximum Independent Set, MIS)的质粒 DNA 计算模型^[4];Liu 等提出了求解 SAT 问题的表面 DNA 计算模型^[5]。

最大匹配问题(Maximum Matching Problem, MMP)是图论中一个经典的组合优化问题,它早已被证明是 NP-完全问题,而且在实践上有着重要的意义,在最优分配、网络优化等方面都有着广泛的应用。针对最大匹配问题,已经有不少研究者提出了解决此问题的 DNA 算法^[6-10],这些算法各有特色。文献^[6]是基于质粒模型求解匹配问题的 DNA 分子算

法,主要通过限制性内切酶的酶切和凝胶电泳完成解的产生和最终解的分离。文献^[7]提出了最大匹配问题的 DNA 表面计算模型,在固体载体上逐步生成解空间的同时,利用酶切技术删除所产生的“不可行解”,从而求得最大匹配。文献^[8]是基于 Adleman 的 DNA 计算模型求解最大匹配问题。上述 3 种算法均涉及到应用限制性内切酶,其数量有限,且酶切操作有时延、易失误,因而不适用于大规模问题的求解。文献^[9]提出了基于粘贴模型求解完美匹配问题的 DNA 算法,其基本思想是通过 PNA 和 DNA 分子之间的特殊链接模式来解决完美匹配问题。此算法仅限于求解完美匹配问题,不适用于求解无完美匹配的最大匹配问题。文献^[10]提出一种基于 Adleman 模型生物操作与粘贴模型解空间的最大匹配问题 DNA 算法,该算法由 4 个子算法组成,能减少最大匹配问题求解空间,而生物操作运算过程较为繁复。

本文提出一种基于粘贴计算模型的求解无向图的最大匹配问题的 DNA 算法。算法使用粘贴计算模型中最基本的生物操作,运算过程中不需要 DNA 链的延伸和酶的作用, DNA 链可重复使用,操作步骤简洁,不会有上述算法的局限性。相比文献^[9]的算法,本文的算法不仅可求解完美匹配问题,也能解决不存在完美匹配解的最大匹配问题,具有普适性。

到稿日期:2013-05-21 返修日期:2013-07-15 本文受国家自然科学基金(61370174),中央高校基本科研业务费专项资金(WH1114030)资助。

吴雪(1957—),女,副教授,主要研究方向为自然计算、通信网系统优化, E-mail: wuxue@ecust.edu.cn;宋晨阳(1987—),男,硕士,主要研究方向为 DNA 计算、模式识别;张楠(1988—),女,硕士生,主要研究方向为 DNA 计算;朱煜(1973—),女,教授,主要研究方向为图像与视频信息处理、模式识别;陈志华(1969—),男,副教授,主要研究方向为图像与视频处理、计算机动画。

2 最大匹配问题及粘贴 DNA 模型

2.1 最大匹配问题描述

在无向图 $G=(V,E)$ 中,对于边集 E 的任意一个子集 $M \subseteq E$,如果 M 中任意两条边在 G 中均不邻接,则称 M 是 G 的一个匹配(matching)。 M 中某条边的两个顶点称为在 M 下是配对的。极大匹配是不能再加入其他边的匹配,一个图可以有许多的极大匹配,其中边数最多的极大匹配称为最大匹配。如图 1 所示, $\{e_1, e_4\}, \{e_2\}, \{e_3\}$ 是图的 3 个极大匹配, $\{e_1, e_4\}$ 是该图的最大匹配。

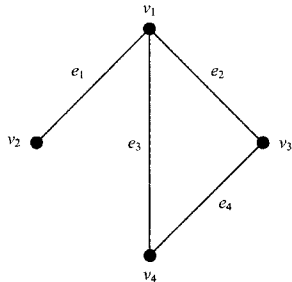


图 1 无向图 G_1 及其最大匹配 $\{e_1, e_4\}$

2.2 粘贴 DNA 计算模型简介

粘贴计算模型是由 Roweis 等人首次提出的^[11]。

粘贴计算模型中的生物操作运算用到的试管是含有 DNA 分子链的集合。粘贴计算模型中的基本操作有 4 种:合并,分离,设置,清除^[12]。

合并(merge):给定两个试管 T_1, T_2 ,定义 $U(T_1, T_2) = T_1 \cup T_2$,即两个试管的存储链合在同一试管,所得到的试管可是两个输入存储链的并集。

分离(separation):对试管中的存储链,根据某一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元上的值为“1”;而另一个试管中存储链在该位元上的值为“0”。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 $+(T, i)$ 所包含的存储链是由试管 T 中所有的第 i 位元值为“1”的存储链构成的,而试管 $-(T, i)$ 所包含的存储链是由试管 T 中所有的第 i 位元值为“0”的存储链构成的。

设置(set):将试管中所有的存储链某位元的值全部转成“1”。即对于试管 T 和整数 $i, 1 \leq i \leq n$,操作设置是产生一个新的试管,使得试管 T 中每个存储链的第 i 个子链转入“开”,即变为双链。对此,如果它的第 i 个子链是“关”的,则需一个相匹配的粘贴链被退火于该存储链的第 i 个子链上;如果第 i 个子链已是“开”状态,则保留原状态。

清除(clear):将试管中所有的存储链中某位元的值全部变为“0”。更详细地讲,对于试管 T 和整数 $i, 1 \leq i \leq n$,清除操作产生一个新的试管,使得该试管 T 中的每个存储链的第 i 个子链转入“关”。最终,由“开”状态加热分离出来的粘贴链被删除。

粘贴计算模型的输入是含有许多 DNA 链的 (K, n) -库的初始试管。一个 (K, n) -库是一个集合,它包含有许多长度为 K 的串,这个 K 值由长度为 n 的所有可能出现的位串以及其余的 $K-n$ 个零值取代而生成。显然,这样在集合共有 2^n 个长度为 K 的串。例如, $(4, 3)$ -库是集合 $\{0000, 0010, 0100, 0110, 1000, 1010, 1100, 1110\}$ 。

对初始试管中有限 DNA 序列进行粘贴计算的生物操作

可构成粘贴 DNA 算法。而根据不同的问题,需设计不同的算法来实现问题的求解。基于粘贴模型已解决的问题有集合覆盖问题、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等^[13]。

3 最大匹配问题的 DNA 算法

3.1 最大匹配问题的初始 DNA 编码

求解最大匹配问题,首先是对于给定的图的各条边进行编码,将其转化为 DNA 序列,从而构建包含解空间的初始试管。因此,解决最大匹配问题的第一步是产生初始试管,在初始试管中包含所有可能的最大匹配。假设一个无向图 $G=(V,E), V$ 是顶点的集合, E 是边的集合。 $|V|=m$ 代表顶点的个数, $|E|=n$ 代表边的条数。设一个 n 位二进制数代表图 G 中一个可能的匹配,边集 $M \subseteq E$ 代表图 G 中的任一个子边集。如果一个 n 位二进制数的第 i 位设置为 1,则它代表相应的第 i 条边在 M 中,而在 $E-M$ (M 的补集)中。如果在一个 n 位二进制数的第 i 位设置为 0,则它代表相应的第 i 条边不在 M 中,而在 $E-M$ 中。这样,所有图 G 中可能边的集合可以转化为 n 位二进制数的集合,如表 1 所列,16 个 2 进制数分别表示了图 1 的各种边的集合。

表 1 二进制数与边集的对应关系

二进制数 X	边集 M
0000	空集
0001	$\{e_1\}$
0010	$\{e_2\}$
0011	$\{e_1, e_2\}$
0100	$\{e_3\}$
0101	$\{e_1, e_3\}$
0110	$\{e_2, e_3\}$
0111	$\{e_1, e_2, e_3\}$
1000	$\{e_4\}$
1001	$\{e_1, e_4\}$
1010	$\{e_2, e_4\}$
1011	$\{e_1, e_2, e_4\}$
1100	$\{e_3, e_4\}$
1101	$\{e_1, e_3, e_4\}$
1110	$\{e_2, e_3, e_4\}$
1111	$\{e_1, e_2, e_3, e_4\}$

这些边的集合并非全部为合法匹配,为了应用 DNA 算法求解合法匹配进而求解最大匹配,需要将这些二进制数编码为 DNA 序列。设一个 n 位二进制数 X_n, X_{n-1}, \dots, X_1 代表一个无符号值 X, X_i 的值是 0 或者 1, $1 \leq i \leq n$ 。 X 的值的范围是从 0 到 2^n-1 ,即 2^n 种可能的值。每个可能的值代表图 G 的任一个子边集。 X 的某一位 X_i 代表 G 的一条边 e_i 。如果第 i 条边在 M 边集中,那么 X_i 的值设置为 1,用 X_i^1 表示;如果第 i 条边不在边集 M 中,则 X_i 的值设置为 0,用 X_i^0 表示。采用文献^[15]中的编码方法,对于图 G_1 中 4 条边的不同情况进行 DNA 序列编码如下:

$X_1^0 = \text{TCTAAGGTTA}$
 $X_1^1 = \text{ATTCACTTCT}$
 $X_2^0 = \text{TACCTCGATT}$
 $X_2^1 = \text{AACATACCCC}$
 $X_3^0 = \text{AACTCGTCAT}$
 $X_3^1 = \text{AAAAC TCACC}$
 $X_4^0 = \text{ATTCTGGCCT}$
 $X_4^1 = \text{CCATATAATC}$

此编码方式使所有的 4 位二进制数均可转化为长度相同的 DNA 序列,如表 2 所列。这样可将二进制数、图子边集

以及 DNA 序列建立一一对应的关系,根据不同的 DNA 序列可以得到其所代表的不同的边集合。所有 DNA 序列亦代表了图 G 所有可能的边集合,即为最大匹配问题解空间的初始编码,进而再对该初始编码解空间进行相应的分子生物操作以求出最大匹配问题的解。

表 2 DNA 序列和二进制的对应关系

DNA 序列	二进制数
ATTCTGGCCTAACTCGTCATTACCTCGATTCTAAGGTTA	0000
ATTCTGGCCTAACTCGTCATTACCTCGATTATTCACCTTCT	0001
ATTCTGGCCTAACTCGTCATAACATACCCCTCTAAGGTTA	0010
ATTCTGGCCTAACTCGTCATAACATACCCCATTCACCTTCT	0011
ATTCTGGCCTAAAACCTACCTACCTCGATTCTAAGGTTA	0100
ATTCTGGCCTAAAACCTACCTACCTCGATTATTCACCTTCT	0101
ATTCTGGCCTAAAACCTACCAACATACCCCTCTAAGGTTA	0110
ATTCTGGCCTAAAACCTACCAACATACCCCATTCACCTTCT	0111
CCATATAATCAACTCGTCATTACCTCGATTCTAAGGTTA	1000
CCATATAATCAACTCGTCATTACCTCGATTATTCACCTTCT	1001
CCATATAATCAACTCGTCATAACATACCCCTCTAAGGTTA	1010
CCATATAATCAACTCGTCATAACATACCCCATTCACCTTCT	1011
CCATATAATCAAAAACCTACCTACCTCGATTATTCACCTTCT	1101
CCATATAATCAAAAACCTACCAACATACCCCTCTAAGGTTA	1110
CCATATAATCAAAAACCTACCAACATACCCCATTCACCTTCT	1111

3.2 求解最大匹配问题的 DNA 算法

本文基于粘贴计算模型提出的求解任一无向图最大匹配问题的 DNA 算法如表 3 所列。

表 3 基于粘贴模型求解 MMP 的 DNA 算法

Algorithm DNA algorithm to solve MMP
(1)input(T_0), T_0 contains all DNA sequence of possible resolution
(2) For all $i=1$ to n , n represents the number of the edge in graph
(a) $T_0 = +(T_0, x_i^1), R = -(T_0, x_i^1)$
(b) For each $j=i+1$ to n, e_j is adjacent to e_i
{
(c) $T_0 = -(T_0, x_j^1), T_s = +(T_0, x_j^1)$
(d) $T_0 = \cup(T_0, R)$
delete T_s
}
(3) For $i=0$ to $n-1$
For $j=i$ to 0
(a) $T_{j+1}^{ON} = +(T_j, x_{j+1}^1), T_j = -(T_j, x_{j+1}^1)$
(b) $T_{j+1} = \cup(T_{j+1}, T_{j+1}^{ON})$
(4) Detect T_k
For $k=n$ to 1
(a) If $T_k \neq \text{null}$
(b) Read(T_k)
END

在步骤(1)中,输入是用 3.1 节所述初始编码产生的一个包含所有可能的 2^n 个 DNA 序列的试管,即初始输入试管中包含了具有 n 条边的无向图的最大匹配问题解空间。

根据最大匹配问题的定义,步骤(2)(a)应用“分离”操作来组建两个试管: T_0 和 R 。第一个试管 T_0 包含所有 $X_i=1$ 的链,第二个试管 R 包括所有 $X_i=0$ 的链, $1 \leq i \leq n$ 。试管 R 代表了集合 $E-E_i$, R 中不包括边 E_i 。对于每一个与 E_i 相关的边 E_j ,继续进行分离操作, T_0 包含所有 $X_j=0$ 的链; T_s 包含所有 $X_j=1$ 的链,即 T_s 中有两条边相邻,为非法匹配。步骤(2)(d)将舍弃试管 T_s ,即 T_s 中所包含的所有的非法最大匹配将被删除;然后将 T_0 和 R 合并成新的 T_0 ,重复以上操作,则所有非法匹配被删除, T_0 中留下的 DNA 链代表了所有合法匹配。

算法每执行步骤(3)的一次外循环,内循环的执行次数为 $(i+1)$ 。首次执行外循环,内循环只执行一次。因此,步骤(3)(a)和(3)(b)也将执行一次。步骤(3)(a)使用“分离”操作

形成两个试管: T_1^{ON} 和 T_0 。第一个试管 T_1^{ON} 包含所有 $X_i=1$ 的链;第二个试管 T_0 包含了所有 $X_i=0$ 的链。试管 T_1^{ON} 包含了所有包括边 E_i 的匹配;试管 T_0 包含了所有不包括边 E_i 的匹配。因此,步骤 3(b)应用“合并”操作将试管 T_1^{ON} 倒入试管 T_1 。经过重复执行步骤(3)(a)和(3)(b),最后产生了 n 个试管。试管 $T_k (n \geq k \geq 1)$ 的编码序列中包含了所有具有 k 条边的匹配。

考虑到问题是要寻找一个最大匹配,步骤(4)(a)执行检测操作,对试管 T_k 进行判断。如果它的返回值非空,那么试管 T_k 包含的匹配是最大匹配,读取试管 T_k 中的 DNA 序列并终止算法。否则,继续重复执行步骤(4)(a)直至最大匹配在试管 T_k 中被检测到。可使用荧光探针的分子信标技术作为 DNA 序列的检测方法。最终满足最大匹配条件的 DNA 序列分离出来后,使其与分子信标部分结合,荧光探针将会读出 DNA 序列。

以图 1 所示的图 G_1 为例,算法的步骤(1)使试管 T_0 充满 16 种 DNA 序列,即 16 种可能的匹配:空集, $\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_2, e_3\}, \{e_2, e_4\}, \{e_3, e_4\}, \{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_1, e_3, e_4\}, \{e_2, e_3, e_4\}$ 和 $\{e_1, e_2, e_3, e_4\}$ 。图 G_1 中边的数量是 4。因此,算法的步骤(2)的执行次数是 6。执行完毕后,生成了试管 T_0 ,包含了合法匹配: $\{e_1\}, \{e_1, e_4\}, \{e_2\}, \{e_3\}, \{e_4\}$ 和空集。这里空集并不是一个合法匹配,但是在步骤(3)中会将其分离,不会影响最后的结果。试管 R 中包含的非法匹配将会被删除。

图 1 的边数 $n=4$,步骤(3)的外循环的执行次数是 4。步骤(3)的内循环的执行次数依赖于外环循环中循环变量的值。反复执行完步骤 3(a)和(3)(b)后,产生了 4 个新的试管,分别是 T_1, T_2, T_3, T_4 。 T_1 包含了仅含 1 条边的匹配: $\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}$; T_2 包含了仅含 2 条边的匹配 $\{e_1, e_4\}$; T_3, T_4 包含的是空集。

执行了步骤(4)(a), T_4, T_3 为空集,不符合条件,检测到 T_2 非空,执行步骤(4)(b),读取试管 T_2 的结果,则图 1 中的最大匹配是 $\{e_1, e_4\}$ 。算法执行完成,得到所求问题的最终解。

3.3 算法的适用性

本文算法适用于求解任一无向图的最大匹配。以下分别对含有完美匹配的无向图和不具有完美匹配的无向图进行算法分析求解,所谓完美匹配是图 G 顶点集 V 中所有顶点均与匹配 M 关联的最大匹配。

图 2 所示的是含有完美匹配的无向图 G_2 ,图 3 是含有奇数顶点的无完美匹配的无向图 G_3 。

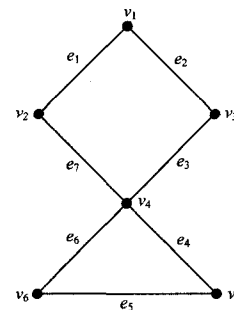


图 2 含有完美匹配的无向图 G_2

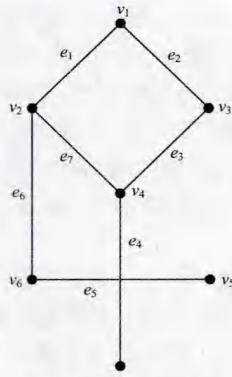


图3 含有奇数顶点的无完美匹配的无向图 G_3

首先对图 G_2 中的 7 条边进行初始编码, 具体编码如下:

- $X_0^1 = \text{GGTAAGGTTA}$
- $X_1^1 = \text{GCTCACTTCT}$
- $X_2^1 = \text{GTCCTCGATT}$
- $X_3^1 = \text{AGCATACCCC}$
- $X_4^1 = \text{CGCTCGTCAT}$
- $X_5^1 = \text{GAAACTCACC}$
- $X_6^1 = \text{TCTAAGGTTA}$
- $X_7^1 = \text{ATTCACCTTCT}$
- $X_8^1 = \text{TACCTCGATT}$
- $X_9^1 = \text{AACATACCCC}$
- $X_{10}^1 = \text{AACTCGTCAT}$
- $X_{11}^1 = \text{AAAACCTCACC}$
- $X_{12}^1 = \text{ATTCTGGCCT}$
- $X_{13}^1 = \text{CCATATAATC}$

然后依据所提出的算法进行生物操作, 每步操作后的结果简要列写如表 4 所列。最终得到图 G_2 的最大匹配为 $\{e_2, e_5, e_7\}$ 和 $\{e_1, e_3, e_5\}$, 这也是图 G_2 的完美匹配。

表 4 求解图 G_2 最大匹配的每步操作结果

步骤(1)	T_0 包含 $2^7=128$ 种全部可能解
步骤(2)	T_0 包含了合法匹配集: $\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_7\};$ $\{e_1, e_3\}, \{e_1, e_4\}, \{e_1, e_5\}, \{e_1, e_6\}, \{e_2, e_7\},$ $\{e_2, e_4\}, \{e_2, e_5\}, \{e_2, e_6\}, \{e_3, e_5\}, \{e_3, e_7\};$ $\{e_2, e_5, e_7\}, \{e_1, e_3, e_5\};$
步骤(3)	形成了 7 个试管 T_1 包含了含有 1 条边的匹配集: $\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_7\};$ T_2 包含了含有 2 条边的匹配集: $\{e_1, e_3\}, \{e_1, e_4\}, \{e_1, e_5\}, \{e_1, e_6\}, \{e_2, e_7\},$ $\{e_2, e_4\}, \{e_2, e_5\}, \{e_2, e_6\}, \{e_3, e_5\}, \{e_5, e_7\};$ T_3 包含了含有 3 条边的匹配集: $\{e_2, e_5, e_7\}, \{e_1, e_3, e_5\};$ T_4, T_5, T_6, T_7 都为 NULL
步骤(4)	从试管 T_7 到 T_1 的顺序开始检测, 将第一个不为 NULL 的试管中的 DNA 序列读出, 得出 $\{e_2, e_5, e_7\}, \{e_1, e_3, e_5\}$ 是该图的最大匹配集

结果表明, 对于同一个具有完美匹配的图, 本文的算法可以得到与文献[9]所提出的求完美匹配算法相同的结果。由图论可知, 图的每个完美匹配都是最大匹配, 反之不真。只有满足 Tutte 充要条件的图才具有完美匹配^[14]。

对于不包含完美匹配的一般无向图中的最大匹配, 本文所提出的算法亦能求得正确解。算法的步骤(3)将包含不同边数的匹配集合分离到 n 个试管 $T_k (n \geq k \geq 1)$ 中, 从而可以通过检测到的非空的最大的下标试管得到包含边数最多的匹配

集, 即无向图的最大匹配。例如, 图 3 所示的图 G_3 含有奇数个顶点, 所以不存在完美匹配, 但是存在最大匹配 $\{e_1, e_3, e_5\}$, $\{e_1, e_4, e_5\}$, $\{e_2, e_4, e_5\}$, $\{e_2, e_4, e_6\}$ 和 $\{e_2, e_5, e_7\}$ 。图 G_3 与图 G_2 同样含有 7 条边, 利用与图 G_2 相同的编码方式, 执行本文的算法, 则可以求得图 G_3 的所有最大匹配。以图 G_1 、图 G_2 和图 G_3 为例的仿真实验结果将在第 4 节中给出。

3.4 算法的复杂度分析

可以证明本文提出的 DNA 算法求解任意 n 条边的无向图的最大匹配问题的复杂度为 $O(n^2)$ 。

除步骤(1)建立初始编码外, 算法主要包含 3 个部分, 其中步骤(2)主要用来确定合法最大匹配解, 删除非法最大匹配解。显而易见, 步骤(2)(a)进行了 n 次“分离”操作。因为每条边最多有 $n-1$ 条邻接的边, 所以步骤(2)(c)最多进行 $n \times (n-1)$ 次“分离”操作, 步骤(2)(d)最多进行 $n \times (n-1)$ 次“合并”操作。步骤(3)是将含有不同边数的最大匹配分离到不同的试管中, 步骤(3)(a)需要 $(n \times (n-1)/2)$ 次的“分离”操作, 步骤(3)(b)需要 $(n \times (n-1)/2)$ 次的“合并”操作。步骤(4)是在合法匹配中确定最大匹配。根据算法, 步骤(4)(a)最多需要 n 次“检测”操作, 步骤(4)(b)只需要一个“读取”操作。由以上分析即可证得本文基于粘贴模型提出的最大匹配问题 DNA 算法的生物操作复杂度为 $O(n^2)$ 。

同样可以得出结论, 运用本文提出的算法求解任意 n 条边的无向图 G 的最大匹配问题的解需要构建 $O(2^n)$ 条初始 DNA 序列, 需要构建 $O(n)$ 个试管, 需要 DNA 序列的编码长度为 $O(10 \times n)$ 。

4 算法的计算机模拟仿真

应用计算机编程语言模拟该 DNA 算法的操作过程, 对 G_1 、 G_2 和 G_3 进行仿真实验的输出结果分别如图 4—图 6 所示。

将图 4 仿真结果 DNA 序列 CCATATAATCAACTCGTCATTACCTCGATTATTCACCTTCT 映射成所对应的边集是 (e_1, e_4) , 即 (e_1, e_4) 是图 G_1 的最大匹配。

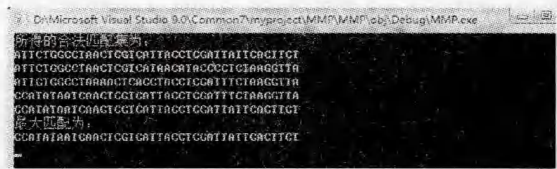


图 4 无向图 G_1 最大匹配问题的仿真结果



图 5 含有完美匹配的无向图 G_2 最大匹配问题的仿真结果

图5所示仿真结果所得DNA序列分别为:ATTCTGGC-CTAACTCGTCATAACATACCCCTCTAAGGTTAGAAA-CTCACCGTCTCGATTGCTCACTTCT,映射成所对应的边集是 (e_1, e_3, e_5) ,这正是图 G_2 的一个最大匹配;

CCATATAATCAACTCGTCATAACATACCCCTCTA-AGGTTACGCTCGTCATAGCATACCCCGGTAAGGTTA,映射成所对应的边集是 (e_2, e_5, e_7) ,这正是图 G_2 的另一个最大匹配。

图6所示仿真结果比图5多以下3个解:

ATTCTGGCCTAACTCGTCATAACATACCCCATTC-ACTTCTCGCTCGTCATGTCCTCGATTGCTCACTTCT,映射成所对应的边集是 (e_1, e_4, e_5) ;

ATTCTGGCCTAACTCGTCATAACATACCCCATTC-ACTTCTCGCTCGTCATGAGCATACCCCGGTAAGGTTA,映射成所对应的边集是 (e_2, e_4, e_5) ;

ATTCTGGCCTAAACTCACCTACCTCGATTATTC-ACTTCTCGCTCGTCATGAGCATACCCCGGTAAGGTTA,映射成所对应的边集是 (e_2, e_4, e_6) 。

边集 $\{e_1, e_3, e_5\}, \{e_1, e_4, e_5\}, \{e_2, e_4, e_5\}, \{e_2, e_4, e_6\}$ 和 $\{e_5, e_7\}$ 正是图 G_3 的所有最大匹配。

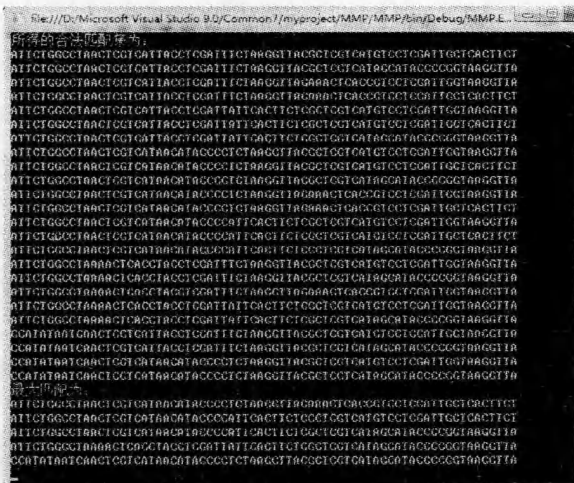


图6 无完美匹配图 G_3 的最大匹配问题的仿真结果

以上3个编程示例表明,仿真结果与预期实验结果完全相同,从而验证了该DNA算法的正确性。

5 算法应用实例

5.1 无线传感器网络移动节点部署问题

无线传感器网络(Wireless Sensor Network)是大量的静止或可移动的传感器节点以自组织或多跳的方式所构成的无线网络,其作用是采集、处理并传输网络覆盖地理区域内感应对象的监测信息,发布给用户。

传感器节点的移动部署的目的是通过将传感器节点移动到指定区域来提高监测目标区域的覆盖率。目标区域可以划分成多个小区域或网格,假设 N 个传感器 S_1, S_2, \dots, S_N 随机部署到 $L \times L$ 的目标区域上,其中 M 个传感器具有移动能力, $N \geq M$ 。设传感器的传感半径为 R ,则部署区域 $L \times L$ 可以划分为面积 $D \times D$ 的网格,其中 $D=R/(\sqrt{2})$,则一共可以形成 $[L/D] \times [L/D]$ 个网格。如果某个网格中至少有一个传

感器,则称此网格被覆盖,否则称此网格为一个空洞。由于能量限制,每个移动节点的移动范围都是有限的,合理地选择移动节点的移动方向成了移动节点部署算法的关键所在。算法的基本思想是,将无线传感器网络的目标区域假设成为一个均匀划分的网格,并将初始网络描述成一个二部图 $G^*=(V, E), V=V_1 \cup V_2$ 。其中, V_1 表示节点集合, V_2 表示需要覆盖的网格集合。如果节点 u 能够覆盖网格 v ,那么边 $(u, v) \in E, u \in V_1, v \in V_2$ 。对于构造的二部图 G^* ,求解出的一个最大匹配便对应着它的一个最优移动部署方案。

图7显示了一个包含有9个节点的传感网络随机部署后的情况。假设部署区域可以划分成 $3 \times 3=9$ 个区域。完成初始部署后共有5个网格 g_1, g_2, g_6, g_8, g_9 被覆盖。假设6个节点 S_1, S_2, \dots, S_6 具有移动能力,且受移动能力限制,每个节点只能移动至上下左右相邻的网格中并且只能移动一次。求节点最佳移动部署方式满足传感器网络的最大覆盖是一个复杂的问题。

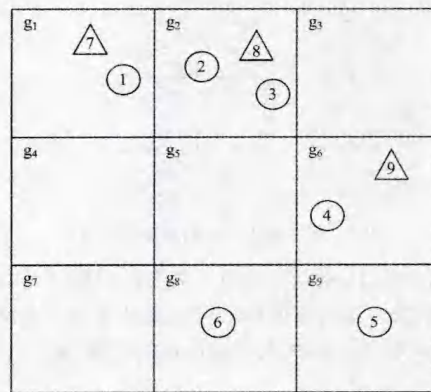


图7 移动节点初始覆盖情况

5.2 移动节点部署的最大匹配问题求解

根据图7中的初始覆盖网络情况,首先将移动节点和目标区域转化成为一个二部图 $G^*=(V, E), V=V_1 \cup V_2$ 。其中, V_1 表示移动节点集合, V_2 表示需要覆盖的区域的集合。具体构造方法如下:

- (1)将所有的可移动节点加入到集合 V_1 中;
- (2)将所有网格加入到集合 V_2 中;
- (3)对于 $\forall u \in V_1, \forall v \in V_2$,如果 u 处于 v 中,或者 u 可达/邻接 v ,则向二部图增加一条有向边 (u, v) 。

则构成如图8所示的初始覆盖对应的二部图 G^* 。

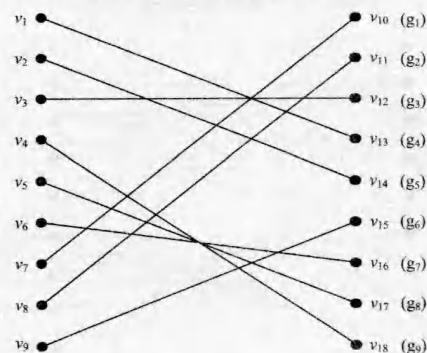


图8 初始覆盖对应的二部图 G

对于二部图 $G^*=(V,E)$, 一个匹配是一个边的集合 $M \subseteq E$, 而且满足对所有的顶点 $v \in V, M$ 中至少有一条边与 v 关联。若 M 中的某条边与 v 相关联, 那么称顶点 v 被匹配。对于图 G^* , 其中任意一条边的一个顶点在移动节点集 V_1 中, 另一个端点目标网格集 V_2 中。显然, 满足最大网络覆盖率的最佳移动节点部署问题就转换成了求解二部图 G^* 的最大匹配问题。

用本文所提出的算法解决此问题可以得到 2 个最大匹配结果: $\{e_{19}, e_{18}, e_{17}, e_{15}, e_{13}, e_{11}, e_7, e_5, e_2\}$ 以及 $\{e_{19}, e_{18}, e_{17}, e_{15}, e_{13}, e_{11}, e_8, e_4, e_2\}$, 如图 9 所示。为了便于观察仿真结果, 在仿真过程中, 将每一位“0”值的 DNA 序列 $X_1^0, X_2^0 \dots X_{19}^0$ 对应成 $a, b \dots s$; 将每一位“1”值的 DNA 序列 $X_1^1, X_2^1 \dots X_{19}^1$ 对应成 $A, B \dots S$ 。



图 9 算法输出结果对应的最大匹配

图 10 和图 11 所示为这两个最大匹配集, 即移动节点部署的最佳方案。通过这两种最佳方案的移动节点部署之后, 网格区域内不再存在空洞, 所有区域被覆盖, 如图 12 和图 13 所示。

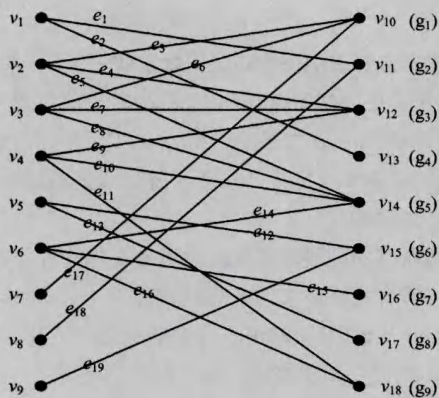


图 10 输出结果对应的最大匹配(1)

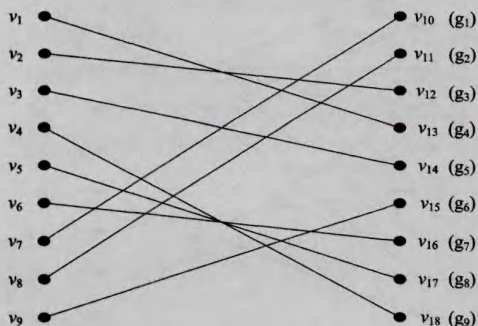


图 11 输出结果对应的最大匹配(2)

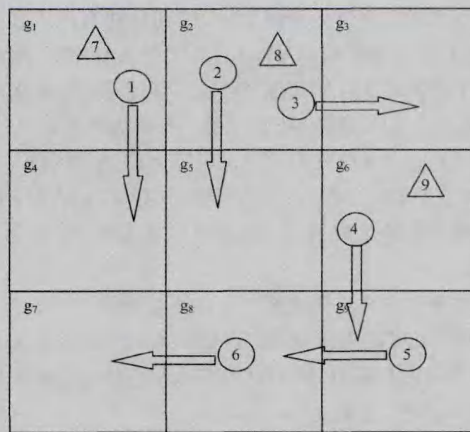


图 12 移动节点部署方案(1)

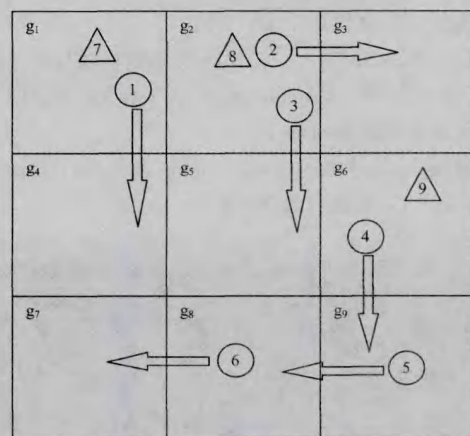


图 13 移动节点部署方案(2)

结束语 DNA 分子计算主要的优势是具有并行运算的能力, 能有效解决图灵机难以解决的某些 NP-完全问题。基于文献[11, 15]的工作, 本文提出最大匹配问题的 DNA 算法。首先对所给图的边用 DNA 序列进行编码, 构建了最大匹配问题的解空间, 然后采用粘贴计算模型中的合并、分离等生物操作对解空间的 DNA 序列进行运算生成可行解, 最后从可行解中分离出符合要求的最大匹配集。计算机仿真实验结果验证了本算法的正确性。

算法所使用的生物操作仅应用了粘贴计算模型中部分简单、快速的生物操作, 且在运算过程中 DNA 链可重复使用, 不需要 DNA 链的延伸和内切酶的作用, 算法运算是一种全自动的生化过程, 因而不但可以提高运算速度, 而且能减少差错产生。

构建的解空间中 2^n 个 DNA 序列正好对应所有可能的 2^n 个子边集合。算法中的步骤(3)可自动将不同边数的匹配分离到 n 个不同的试管, 这样在得到最大匹配的同时, 也取得了其他任意匹配的信息, 因而适用性强。

本算法初始试管中最长 DNA 链的长度和所需 DNA 链的数量分别为 $O(10 \times n)$ 和 $O(2^n)$, 这说明随着边的数量的增加, DNA 链的长度和数量都会大幅增加, 则在生化反应过程中, 生物操作可能产生差错的概率将会增加, 从而计算的成本也将会增加。如何减小最大匹配问题解空间的规模是需要我们进一步研究解决的。

(下转第 140 页)

实验结果中性质标识 f_1, f_2 出现在回答集中,说明对于 PHS 并发系统,性质 1 与性质 2 是可满足的,与事实一致。

以上实例说明本文建立的 ASP 的模型检测框架以及并发系统性质验证方法是可行的、有效的,能够验证多个(≥ 3)进程并发组合的模型,并一次验证多条性质。

结束语 本文研究了基于 ASP 的 CSP 进程描述与组合生成技术,构建了并发系统的 ASP 建模与验证框架,完善了可描述并发进程形态,扩大了可验证并发系统规模。实验结果表明了本文构建的 ASP 描述体系、并发组合进程生成技术以及基于该 ASP 描述体系的性质验证的有效性。然而随着系统规模的扩大,面对状态爆炸的难题,如何针对特定类型的待验证性质,对 ASP 构建的模型进行有效的抽象约简,提高性质验证效率,是下一阶段的研究重点,同时性质验证不满足时的反例生成技术也是研究点之一。

参考文献

- [1] Hoare C A R. Communicating Sequential Processes [M]. <http://www.usingcsp.com/cspbooks>, 2004
- [2] Roscoe A W. The Theory and Practice of Concurrency [M]. Prentice Hall, 1998
- [3] Clarke E M, Grumberg O, Peled D. Model Checking [M]. Cambridge, MIT Press, 2001; 35-49
- [4] Baral C. Knowledge Representation, Reasoning, and Declarative Problem Solving [M]. Cambridge Press, 2003
- [5] 赵岭忠, 张超, 钱俊彦. 基于 ASP 的 CSP 并发系统验证研究[J]. 计算机科学, 2012, 39(12): 133-136
- [6] Durobvin J. Efficient Symbolic Model Checking of Concurrent

System [D]. Aalto University School of Science Department of Information of Computer Science, Doctoral Dissertation, 2011

- [7] D'Silva V, Kroening D, Weissenbacher G. A Survey of Automated Techniques for Formal Software Verification [J]. IEEE Trans. on CAD of Integrated Circuits and Systems, 2008, 27(7): 1165-1178
- [8] Heljanko K, Niemelä I. Bounded LTL model checking with stable models [J]. TPLP, 2003(4/5): 519-550
- [9] Armstrong P J, Goldsmith M, Lowe G, et al. Recent Developments in FDR [J]. CAV, 2012, 7358: 699-704
- [10] Palikareva H, Ouaknine J, Roscoe B. Faster FDR Counterexample Generation Using SAT-Solving [C]//ECEASST. 2009
- [11] Leone N, Pfeifer G, Faber W, et al. The DLV system for knowledge representation and reasoning [J]. ACM TOCL, 2006, 7(3): 499-562
- [12] Lierler Y, Maratea M. Cmodels-2: SAT-Based Answer Set Solver Enhanced to Non-tight Programs [J]. Proc. LPNMR, 2004, 2923: 346-350
- [13] Ilik D, Lee G, Herbelin H. Kripke models for classical logic [J]. Ann. Pure Appl. Logic, 2010, 161(11): 1367-1378
- [14] De Angelis E, Pettorossi A, Proietti M. Synthesizing Concurrent Programs Using Answer Set Programming [J]. Fundam. Inform, 2012, 120(3/4): 205-229
- [15] 蒋屹新, 林闯, 曲扬, 等. 基于 Petri 网的模型检测研究[J]. 软件学报, 2004, 15(9): 1265-1276
- [16] Niemelä I, Simons P, Syrjänen T. Smodels, A System for Answer Set Programming [C]// Proceeding of the 8th International Workshop on Non-Monotonic Reasoning. 2000

(上接第 132 页)

参考文献

- [1] Adleman L. Molecular computation of solutions to combinatorial problems[J]. Science, 1994, 266(11): 1021-1024
- [2] Lipton R J. DNA solution of computation problems [J]. Science, 1995, 268(4): 542-545
- [3] Quyang Q, Kaplan P D, Liu Shu-mo, et al. DNA solution of maximal clique problem[J]. Science, 1997, 278(17): 446-449
- [4] Head T, Rozenberg G, Bladergroen R R, et al. Computing with DNA by operating on plasmids[J]. Biosystem, 2000, 57: 87-93
- [5] Liu Q H, Wang L, Anthony G F, et al. DNA computing on surface[J]. Nature, 2000, 403(13): 175-179
- [6] 高琳, 马润年, 许进. 基于质粒 DNA 匹配问题的分子算法[J]. 生物化学与生物物理进展, 2002, 29(5): 820-823
- [7] 刘文斌, 高琳, 王淑栋, 等. 最大匹配问题的 DNA 表面计算模型[J]. 电子学报, 2003, 31(10): 1496-1499
- [8] 谢飞舟, 汤建钢. 最大匹配问题的 DNA 试管计算模型[J]. 甘肃联合大学学报, 2012, 26(6): 65-68

- [9] Dong Ya-fei, Zheng Xue-dong. Molecule Algorithm for Perfect Matching Problem Based on Sticker Models[C]//Proceeding of International Conference on Intelligent Mechatronics and Automation. China, 2004: 249-253
- [10] 周旭, 李肯立, 乐光学, 等. 一种最大匹配问题 DNA 计算算法[J]. 计算机研究与发展, 2011, 48(11): 2147-2154
- [11] Roweis S, Winfree E, Burgoyne R, et al. A sticker-based model for DNA computation[J]. Comput Biol, 1998, 5(4): 615-629
- [12] 许进, 董亚非. 粘贴 DNA 计算机模型(I): 理论[J]. 计算机学报, 2004, 49(3): 205-212
- [13] 许进, 董亚非. 粘贴 DNA 计算机模型(II): 理论[J]. 计算机学报, 2004, 49(4): 299-307
- [14] Bondy J A, Murty M S. Graph Theory with Applications[M]. The Macmillan Press, LTD, 1976
- [15] Braich R S, Johnson C, Rothmund P W K, et al. Solution of a satisfiability problem on a gel-based DNA computer, 2000 [C]// Proceedings of the 6th International Conference on DNA Computation in the Springer-Verlag Lecture Notes in Computer Science series. 2000, 2054: 27-42