

求解 0-1 规划问题的 DNA 计算模型

强小利¹⁾ 曾 波¹⁾ 王子成¹⁾ 寇 铮²⁾

¹⁾(华中科技大学控制科学与工程系 武汉 430074)

²⁾(中国科学院武汉病毒研究所病毒学国家重点实验室 武汉 430071)

摘 要 DNA 计算是以 DNA 分子作为数据的一种新型计算模式. 在 DNA 计算中首要面对的问题是编码问题. 文中提出了一种双编码方法, 利用这种编码方法可以使得在 DNA 计算的读解过程类似于 DNA 测序过程, 容易实现自动化操作. 基于该编码方法所建立的 DNA 计算模型可用于求解 0-1 规划问题, 只需 4 次 PCR 反应即可读取问题的可行解. 与其他 DNA 计算模型相比, 该模型具有操作简单、易于实现的优点.

关键词 DNA 计算; 0-1 规划问题; 编码

中图法分类号 TP301

A DNA Computation Model to Solve 0-1 Programming Problem

QIANG Xiao-Li¹⁾ ZENG Bo¹⁾ WANG Zi-Cheng¹⁾ KOU Zheng²⁾

¹⁾(Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074)

²⁾(State Key Laboratory of Virology, Wuhan Institute of Virology, Chinese Academy of Sciences, Wuhan 430071)

Abstract DNA computing is a novel computation paradigm with DNA molecules as ‘data’, and encoding is a crucial problem with great difficulties of DNA computing. In this paper, a novel encoding method named double encoding method is proposed, which could make the procedure of solution detection similar to DNA sequencing technology. By using this method a DNA algorithm to solve 0-1 programming problem is proposed and PCR is done only 4 times of to detect the feasible solutions. Compared with other DNA computing algorithms, this method could be easier and faster to read out the solution.

Keywords DNA computing; 0-1 programming problem; encoding

1 Introduction

A DNA-based polynomial-time method to solve Hamiltonian path problem (HPP) was reported by Adleman in 1994^[1]. The result showed that parallelism of DNA allows DNA computers to solve painstaking problems such as NP-complete problems with linearly increasing time. Since that, the researches in recent years confirmed the above view point. Such as, in 1995, Lipton proposed

molecular biology experiments to solve the 3-SAT problem^[2]; the surface-based DNA computation was used to solve SAT problem^[3] in 2000; the group led by Shapiro made an automatic DNA computer model for diagnosing and curing diseases^[4-5]; and in 2002, Braich solved a 20-variable instance of the 3-satisfiability (3-SAT) problem by DNA computing, where the candidate solutions were 1.04 million, which is, until now, the largest instance solved by non-electronic computer^[6].

收稿日期: 2008-04-20; 最终修改稿收到日期: 2008-10-28. 本课题得到国家自然科学基金(60533010, 30670540, 60874036, 60503002)、国家“八六三”高技术研究发展计划项目基金(2006AA01Z104)、中国教育部博士点基金(20070001020)和中国博士后科学基金(20060400344)资助. 强小利, 女, 1979 年生, 博士研究生, 目前主要从事分子计算模型的理论和应用、生物信息学等方面的研究. E-mail: xliqiang@mail.hust.edu.cn. 曾 波, 男, 1986 年生, 硕士研究生, 研究方向为分子计算、膜计算. 王子成, 男, 1976 年生, 博士研究生, 研究方向为分子计算. 寇 铮, 男, 1979 年生, 博士, 研究方向为生物信息学、分子病毒学与进化.

DNA sequences design is a crucial problem for DNA computing and many algorithms were used to encode. For example, Feldkamp^[7] designed a sequence generation based on the uniqueness of overlapping subsequences; Kobayashi^[8] presented the template method to design DNA sequences by using the Hamming distance to avoid undesired hybridization reaction; the Euler Graph was used by Błewicz^[9] to design DNA sequences; and in 2005, Tanaka^[10] designed software to get codes sequences by calculating free energy (ΔG). Sum up many factors, such as ΔG , T_m , enzymes, the Hamming distance and the composition of DNA molecules, were considered to decrease the nonspecific hybridization^[11]. Nearly each program of sequences generation was aimed on the equilibrium of the constraints and the number of practicable DNA sequences. However, the number of molecules required for solution grows exponentially, which will limit the maximal size of the problem to be solved^[12]. Sometimes to get adequate practicable DNA sequences, some constraints should be neglected, which would make the molecular biology operation imprecise^[13].

In the paper, a DNA computation model is proposed to solve 0-1 programming problem according to the experiment done by Ouyang^[14]. This model attempts to improve the feasibility and validity of biochemistry operation by using a small quantity of DNA sequences. Once these sequences are designed, they could be used for other 0-1 programming problems with n variable. Further more the procedure of detecting solutions is somewhat like DNA sequencing procedure^[15], which could make the DNA computation be autoimmunization.

2 0-1 Programming Problem

The 0-1 programming problem is a special form of integer programming problem, in which the value of variable x_i can only be 0 or 1. It is an important problem applied widely in engineering. There are many algorithms to solve this problem, such as invisible enumeration method and exhaust algorithm.

The special form of 0-1 programming problem which we propose to solve in the paper is

$$\begin{aligned} \max(\min) z &= c_1x_1 + c_2x_2 + \cdots + c_nx_n, \\ \text{s. t. } \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq (=, \geq) b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq (=, \geq) b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq (=, \geq) b_m \end{cases} \\ x_1, x_2, \cdots, x_n &= 0, 1 \end{aligned}$$

In the paper the 0-1 programming problem is as same as reference [16].

$$\begin{aligned} \min u &= 2x + y + 3z, \\ \text{s. t. } \begin{cases} x + z \leq 1 \\ x + y + z \geq 2 \\ y + z \leq 1 \\ x, y, z = 0, 1 \end{cases} \end{aligned}$$

3 DNA Computation Model for 0-1 Programming Problem

3.1 Description of the Problem

The DNA algorithm was designed as followed:

Step 1. Synthesize all DNA strands representing all possible solutions.

Step 2. Delete unfeasible solutions through digestion reaction by specific restriction endonucleases.

Step 3. Detect and read out the feasible solutions by polymerase chain reaction (PCR).

3.2 Double Encoding Method

Two sets of codes were used. One is specific sequences of restriction endonuclease as different values for each variable, so the DNA strands representing the unsatisfied solutions should be digested by the restriction endonucleases. The other set codes are 6 DNA sequences which all has 15 bases. By using the second set codes, PCR could be done to readout the satisfied solutions.

3.2.1 Encoding for variable

The existing specific recognition sites of restriction endonucleases were used to denote the different values for different variable. The length of these DNA sequences is 6 bases.

The codes for values of variable were chosen from existing DNA sequences of restriction endonucleases. The constraints for sequence choice are list as below: (1) the digestion reaction should be high efficiency; (2) the isoschizomer should not be used; (3) all endonucleases buffers should be identical biologically.

For a variable x with two values 0 or 1, two different DNA sequences could be used to represent 0 or 1 respectively. According to the constraints above, we choose 6 sequences (Table 1) to represent the values of variable x, y, z . Based on this set code, the digestion reaction should be done.

If there are n variables, $2n$ DNA sequences could be used, that is $2n$ restriction endonucleases could be used.

Table 1 The Endonucleases for Values of Variables

Restriction endonucleases		DNA sequences
x^0	SmaI	CCCGGG
		GGGCCC
x^1	Aat II	GACGTC
		CTACAG
y^0	Alw44I	GTGCAC
		CACGTG
y^1	BauI	CACGAG
		GAGCAC
z^0	Bsu15I	ATCGAT
		TAGCTA
z^1	Bsp1407I	TGTACA
		ACATGT

3.2.2 Encoding for linker

These codes include 6 DNA sequences of 15 bases separately, which could be divided into two set: primer sequences and linker sequences.

Primer sequences; P_1 and P_2 were used to denote the beginning and the end sequences of all DNA strands representing all possible solutions, and used as primer pair to PCR for amplifying DNA strands. ‘Linker’ sequences, denoted as L_j , $j = 1, 2, 3, 4$, are between any two variable sequences. Their first part represents the value of variable i and their second part represents the value of variable $i + 1$. The combination of L_j is showed in Table 2. For example if L_2 is between x_i and x_{i+1} , the DNA sequence could be like $x_i^0L_2x_{i+1}^1$. These ‘linker’ sequences contain not only the position information but also the information of variables.

Table 2 Combination of L_j

	Linker	
	0	1
0	L_1	L_2
1	L_3	L_4

Note: The letters in column 1 (left) represent the value of variable i and the letters in row 1 (above) represent the value of variable $i + 1$.

To get these 6 sequences (Table 3), a simple program was designed according to several constraints.

Table 3 DNA Sequences for L_j

Linker	DNA sequences
L_1	5'-AAACTTCCTTCACCC-3'
L_2	5'-CTCACACTTCTCCTA-3'
L_3	5'-CTAAACATCCCCTAC-3'
L_4	5'-TCACTCAACACTCAC-3'
P_1	5'-CAACCATCTCTACTC-3'
P_2	5'-ACCCCTTCTATCACA-3'

(1) Each DNA sequence is 15 base long only containing A’s, T’s and C’s.

- (2) Each DNA sequence has no occurrence of 5 or more consecutive identical bases.
- (3) Every sequence has 7 or 8 or 9 Cs.
- (4) The length of unique overlapping subsequences is not more than 5 bases.
- (5) The Hamming distance of every sequence is more than 7.

Once these 6 sequences are confirmed, they could be used for other 0-1 programming problems with n variables.

3.3 Construction for All Possible Solutions

The alternative array of x_i and L_j could construct all the DNA strands representing all possible solutions. The set of all DNA strands is

$$S = \{P_1x_1L_jx_2L_j \cdots x_nP_2; j = 1, 2, 3, 4\}.$$

Where x_i is arrayed alphabetically, but the position of L_j on a DNA strand is not stochastic. It must be corresponding with the values of variable closed. For example, $x_i^0L_2x_{i+1}^1$ could be synthesized, but $x_i^0L_1x_{i+1}^1$ must be avoided. So the method to synthesize the DNA strands of S is important.

DNA strands representing all possible solutions is synthesized alphabetically according to variables, and each DNA strand includes 3 sequences of restriction endonucleases site, 2 linker sequences, and P_1, P_2 . The length of restriction endonuclease sequence is 6 base pairs (bp), and other sequences are 15 bp. Therefore, every DNA strands have 78 bp. The primer pair $\langle P_1, \overline{P_2} \rangle$ is used to amplify the DNA strands (data pool).

3.4 Delete Unfeasible Solutions

In the paper, restriction endonucleases were used to digest the DNA strands representing unfeasible solutions according to the constraints (Fig. 1). For the first constraint $1, x + z \leq 1$, means the values variable x and z could not be 1 simultaneous. To implement this constraint, the data pool was divided into two test tube: T_1 and T_2 . In the tube T_1 , the restriction endonucleases AatII was added to delete the sequences with x^1 , and the restriction endonucleases Bsp1407I was used to delete the sequences with z^1 . Then this two tubes were mixed into a new tube T , and the DNA strands with x^1z^1 simultaneous should be cut.

For the constraint $2, x + y + z \geq 2$, and constraint $3, y + z \leq 1$, the computational process would be done in the same way. Notice that, to delete $x^0y^0z^0$, 3 tubes should be used. The specific operation has an analogy to the reference [14].

After digestion reaction, the product was used to PCR to read out the feasible solutions, and only the DNA strand with $x^1y^1z^0$ could be preserved.

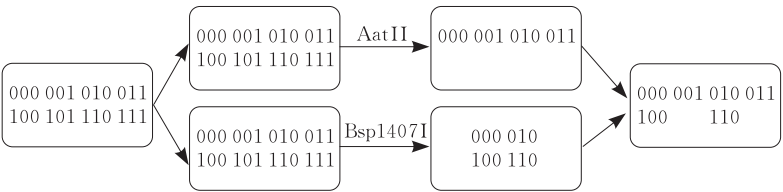


Fig. 1 Deletion of unfeasible solutions

3.5 Detect and Read Out the Feasible Solutions by PCR

To readout the feasible solutions, $\langle P_1, \overline{L_j} \rangle$ was used as primer pares for PCR to read out the feasible solution. We define: \emptyset denotes that there is no product after PCR. F_i means that some DNA fragments are produced after PCR, and the primer used for this product is L_j after variable x_i .

The products of PCR could be DNA ladder in the gel. According to the length of these fragments, the position of L_j could be judged, and the values of the variables connecting with L_j could also be determined. The reaction numbers of PCR were only 4 times for any problem with n variable.

In the paper, two DNA fragment could be get in the gel, and the shorter is 36 bp, the longer is 57 bp. For 36 bp, L_j is after variable x , and for 57 bp, L_j is after variable y . The products are list in Table 4.

Table 4 Detection of the Satisfied Solutions

Primer pairs	Products of PCR	DNA sequences without linkers
$\langle P_1, \overline{L_1} \rangle$	\emptyset	
$\langle P_1, \overline{L_2} \rangle$	\emptyset	
$\langle P_1, \overline{L_3} \rangle$	57 bp	F_2 $y^1 z^0$
$\langle P_1, \overline{L_4} \rangle$	36 bp	F_1 $x^1 y^1$

It is easy to know that two DNA fragment $(y^1 z^0, x^1 y^1)$ has same portion (y^1) , so these two sequences could be connected to form $x^1 y^1 z^0$, which is the feasible solution for the problem. This procedure of detecting solutions is somewhat like sequencing procedure.

4 Discussion

DNA computing is large-scale parallel calculation with enormous information storage. The most difficult problem to obstruct the development of DNA computing is the encoding problem. To get adequate DNA sequences, some constraints should be neglected, which could decrease the reliability of the experiment.

In the paper two groups of codes were used for digestion reaction and PCR, and the reaction time was less than other DNA algorithm. Especially the

PCR was only four times for any 0-1 programming problems with n variables. Compared with other methods of DNA algorithm, fewer biological experiments could be done and the calculation error could be less too.

After PCR, DNA ladder fragments could be obtained. Using these fragments, the products could be connected to DNA strands, which represent the feasible solutions. Briefly, the procedure of detecting solutions is somewhat like sequencing procedure. With digestion reaction manipulator and microflux control applied, roboticized operation of DNA computing could be realized. But, $2n$ restriction endonucleases should be used for a problem with n variables. How to reduce the number of endonucleases is still a problem.

References

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

[2] Lipton R J. DNA solution of hard computational problems. Science, 1995, 268: 542-544

[3] Liu Q H, Wang L M, Fruto A G. DNA computing on surface. Nature, 2000, 403: 75-179

[4] Benenson Y, Paz-Elizur T, Adar R, Keinan E, Livneh Z, Shapiro E. Programmable and autonomous computing machine made of biomolecules. Nature, 2001, 414: 430-434

[5] Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. Nature, 2004, 429: 423-429

[6] Braich R S, Chelyapov N, Johnson C, Rothmund P W K, Adleman L M. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296: 499-502

[7] Feldkamp U, Rauhe H, Banzhaf W. Software tools for DNA sequence design. Genetic Programming and Evolvable Machines, 2003, 4: 153-171

[8] Kobayashi S, Kondo T, Arita M. On template method for DNA sequence design. Letter Notes in Computer Science, 2003, 2568: 205-214

[9] Bl wicz J, Formanowicz P, Kasprzak M, Schuurman P, Woeginger G J. DNA sequencing, eulerian graphs, and the exact perfect matching problem. Lecture Notes in Computer Science 2573, London: Springer-Verlag, 2002: 13-24

[10] Tanaka F, Kameda A, Yamamoto M, Ohuchi A. Design of nucleic acid sequences for DNA computing based on a thermodynamic approach. Nucleic Acids Research, 2005, 33(3): 903-911

[11] Tulpan D, Andronescu M, Chang S B, Shortreed M R, Con-
don A, Hoos H H, Smith L M. Thermodynamically based
DNA strand design. *Nucleic Acids Research*, 2005, 33(15):
4951-4964

[12] Aoi Y, Yoshinobu T, Tanizawa K, Kinoshita K, Iwasaki
H. Ligation errors in DNA computing. *BioSystems*, 1999,
52: 181-187

[13] Liu W B, Wang S D, Xu J. Research on the encoding method
of DNA computing. *Computer Engineering and Applications*,
2003, 27: 118-121

[14] Ouyang Q, Kaplan P D, Liu S, Libchaber A. DNA solution
of the maximal clique problem. *Science*, 1997, 278: 446-449

[15] Sambrook J, Russell D W. *Molecular Cloning: A Laboratory
Manual*. 3rd Edition. New York: Cold Spring Harbor Labo-
ratory Press, 2001

[16] Zhang F Y, Yin Z X, Liu B, Xu J. DNA computation model
to solve 0-1 programming problem. *Biosystem*, 2004, 74:
9-14



QIANG Xiao-Li, born in 1979,
Ph. D. candidate. Her current research
interests are biomolecular computing,
bioinformatics etc.

ZENG Bo, born in 1986, M. S. candidate. His main re-
search interests are biomolecular computing and membrane
computing.

WANG Zi-Cheng, born in 1976, Ph. D. candidate. His
current research interests focus on biomolecular computing.

KOU Zheng, born in 1979, Ph. D.. His current re-
search interests are bioinformatics, molecular virology and
evolution.