

Internet Electronic Journal of Molecular Design

May 2004, Volume 3, Number 5, Pages 287–294

Editor: Ovidiu Ivanciuc

Special issue dedicated to Professor Nenad Trinajstić on the occasion of the 65th birthday
Part 11

Guest Editor: Douglas J. Klein

A Surface–Based DNA Computing for the Simple 0–1 Programming Problem

Zhi–xiang Yin,^{1,2} Jia–xiu Zhang,¹ and Jin Xu²

¹ Department of Mathematics and Physics, Anhui University of Science and Technology, Anhui
Huainan (232001) China

² Department of Control Science and Engineering, Huazhong University of Science and
Technology, Hubei Wuhan (430074) China

Received: February 18, 2003; Revised: June 2, 2003; Accepted: September 29, 2003; Published: May 31, 2004

Citation of the article:

Z. Yin, J. Zhang, and J. Xu, A Surface–Based DNA Computing for the Simple 0–1 Programming
Problem, *Internet Electron. J. Mol. Des.* 2004, 3, 287–294, <http://www.biochempress.com>.

A Surface–Based DNA Computing for the Simple 0–1 Programming Problem[#]

Zhi–xiang Yin,^{1,2,*} Jia–xiu Zhang,¹ and Jin Xu²

¹ Department of Mathematics and Physics, Anhui University of Science and Technology, Anhui Huainan (232001) China

² Department of Control Science and Engineering, Huazhong University of Science and Technology, Hubei Wuhan (430074) China

Received: February 18, 2003; Revised: June 2, 2003; Accepted: September 29, 2003; Published: May 31, 2004

Internet Electron. J. Mol. Des. 2004, 3 (5), 287–294

Abstract

DNA computing is a novel method of solving a class of intractable computational problem, in which the computing speeds up exponentially with problem size. Up to now, many accomplishments have been made to improve its performance and increase its reliability. In the paper, we solved the simple 0–1 programming problem with fluorescence labeling techniques based on surface chemistry by attempted to apply DNA computing to programming problem. Our method has some significant advantages such as simple encoding, low cost, and short operating time.

Keywords. DNA computing; 0–1 programming problem; NP–complete problem.

Abbreviations and notations

DHPP, Directed Hamiltonian Path Problem

SAT, The Satisfiability Problem

1 INTRODUCTION

In 1961, Feynman gave a visionary talk describing the possibility of building computers that were sub–microscopic [1]. Despite remarkable progress in computer miniaturization, this goal has yet to be achieved. Computer scientists rank computational problems in three classes: easy, hard and incomputable [2]. About thirty years ago there was developed a conception designating a hierarchy of complexity classes for problems on finite sets. And so long as we use digital computers with finite memory storing discrete objects to resolve computational problems, it is relevant for any non–trivial algorithm designing. With the current state–of–the–art the most important complexity classes are P (problems solvable in polynomial time) and NP (problems whose solution certificate can be verified in polynomial time). The most fruitful result of the conception is that complexity

[#] Dedicated to Professor Nenad Trinajstić on the occasion of the 65th birthday.

* Correspondence author; phone: 86–27–87543563; fax: 86–27–87543130; E–mail: zhixiangyin@sohu.com.

classes have so-called complete problems. A problem of a class is complete if you can solve any other problem of this class in polynomial time having a polynomial time algorithm for the first one. Hence complete problems are hardest in their own classes and as they exist we may choose any of them to advance solving techniques for the entire class. The concept of complete problems for a class is generalized to hard problems for the class by inclusion of all other problems, whose polynomial time algorithm gives polynomial time solvability for the class. So, there are NP-complete and NP-hard problems [3,4]. One of the major achievements of computer science in the last two decades is the understanding that many important computational search problems are NP-complete and thus are unlikely to have efficient algorithms that solve the problem exactly. Adleman (1994) showed that DNA can be used to solve a computationally hard problem, the directed hamiltonian path problem (DHPP), and demonstrated the potential power of parallel, high-density computation by molecules in solution [5]. This parallelism allows DNA computers to solve larger hard problems such as NP-complete problems in linearly increasing time, in contrast to the exponentially increasing time required by an electronically computer. After Adleman initiated the field of DNA computing in 1994, Lipton (1995) proposed DNA experiments to solve the satisfiability (SAT) problem [6]. In 1997, Ouyang *et al.* presented a molecular biology based experimental solution to the "maximal clique" problem [2]. In 2000, Liu *et al.* designed DNA model system; a multi-based encoding strategy is used in a one-word approach to surface-based DNA computation [7]. In 2001, Wu analyzed and improved their surface-based method [8]. In 2002, Yin *et al.* gave a Chinese postman problem based on DNA computing [9]. All of these efforts made use of molecular biology and demonstrated the feasibility of carrying out computation at the molecular level. One of the formal frameworks for molecular computations is Tom Head's splicing system, which gives a theoretical foundation for computing based on DNA recombination [10]. 0-1 programming problem and the satisfiability problem are mutually related closely, and 0-1 programming problem is a generalization of the satisfiability problem. Up to now, there have been many results for solving the satisfiability problem [6,7,11,12]. In 2002, Braich *et al.* solved a 20-variable instance of the NP-Complete three-satisfiability problem on a simple DNA computer, and proposed this computational problem may be largest yet solved by nonelectronic means [13]. However, the model of 0-1 programming problem based on DNA computing has never been studied. In this paper, the simple form of the 0-1 programming problem was solved by fluorescence labelling techniques based on surface chemistry. Despite significant progress, several problems remain and need to be resolved. The first, for a complex issue, there is a need of a great amount of DNA in coding, which is hard to be achieved. Secondly, DNA computing is inaccurate, which can be caused by inaccurate hybridization, the effect of secondary structure of DNA molecule, the inaccuracy of experiment and large cost for biological lab experiments, all of these can affect the result of DNA computing. For terminologies and notations not defined in this paper, the readers are referred to Ref. [14].

Certain oligonucleotides are tagged with fluorescent as DNA probes;

Step 2: For each inequality, adding the corresponding complementary strand to the surface, any solution, which satisfies this inequality will be hybridized at least (not exceeding) b_i complementary strand tagged with a fluorescent label. Further, we can determine the solution that satisfy (or does not satisfy) constraint conditions by a method of fluorescent–image;

Step 3: The temperature is raised to separate all double–stranded DNA into single–strands by thermal denaturation. The surface is returned to the initial state by washing in buffer (without regard for infeasible solution determined in step 2);

Step 4: Repeat step 2, step 3, we can reject all infeasible solution and obtain feasible solution of the problem;

Step 5: By calculating and comparing the value of the object function corresponding to every feasible solution, an optimum solution can be obtained.

Remark:

When a_{ij} is real number, both sides of the inequality are number of times to make every coefficient of a variable become an integer. When a_{ij} is an integer, $a_{ij}x_j$ is represented as $x_j + x_j + \dots + x_j$ (where the number of x_j is $|a_{ij}|$).

3. THE 0–1 PROGRAMMING PROBLEM MODEL SYSTEM

For a system of equations that contains n variables x_1, x_2, \dots, x_n and m equations, in order to implement step 1 of biological algorithm mentioned above, two steps are needed. One of them is to synthesis $3n$ oligonucleotides divided into 3 groups, which include n oligonucleotides in each group. The oligonucleotides in the first group represent variable x_1, x_2, \dots, x_n respectively; the second represent variable $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ respectively ($x_i = 1$ if and only if $\bar{x}_i = 0$); the third group represent complementary strands of the first group respectively, individual written as x_1, x_2, \dots, x_n . We elect oligonucleotides x_1, x_2, \dots, x_n and $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ that they must be very different, and for evading misfit among them, at least 4 bp (base pairs) are diversity in oligonucleotides sequence (pay attention to oligonucleotide x_i represents variable $x_i = 1$ and oligonucleotide \bar{x}_i represents variable $\bar{x}_i = 0$). The other is to structure DNA probe with the former two group $2n$ oligonucleotides. The progress can be separated into five steps:

Constructing $2n$ probes corresponding to above single stranded DNA molecules (oligonucleotide) and tagging oligonucleotides x_1, x_2, \dots, x_n with fluorescent; (2) fix untagged DNA strand on the surface by means of six to nine atoms where the DNA stands are arranged in 2^n rows representing all variables of the given computational problem. (3) To implement step 2, add the complementary strand corresponding to each variable of constraint equation to the surface. Any

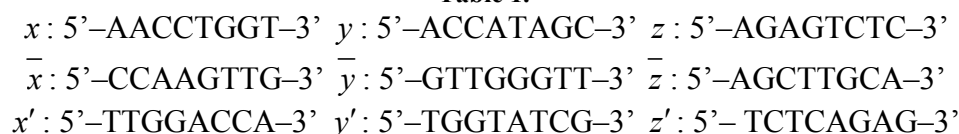
solution satisfies this inequality will be hybridized at least (not exceeding) b_i complementary strand tagged with a fluorescent label. Further, we can determine the solution of satisfy (or does not satisfy) constraint equation by a method of fluorescence–image, and observe their color and record; (4) To implement step 3, the temperature is raised to separate all double–stranded DNA into single–strands by thermal denaturation. The surface is returned to the initial state by washing in buffer (without regard for infeasible solution determined above); (5) To implement step 4, we can reject all infeasible solution and obtain feasible solution of the problem by repeating (2), (3); To implement step 5, comparing to value of object function corresponding every feasible solution, we can obtain optimum solution. We discuss in detail simple 0–1 programming problem as below:

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

To discuss the 0–1 programming problem, the progress was separated into six steps:

(1) We first synthese 9 oligonucleotides divided into the same 3 groups. 3 oligonucleotides of the first group represent variable x, y, z respectively; ones of second group represent variable $\bar{x}, \bar{y}, \bar{z}$ respectively ($x = 1$ if and only if $\bar{x} = 0$, such as y, z); ones of third group represent complementary strand of the first group respectively, individually written as x', y', z' (see Table1) (pay attention to oligonucleotide x represents variable $x = 1$ and oligonucleotide \bar{x} represents variable $\bar{x} = 0$, y and z are also so).

Table 1.



(2) Then we structure DNA probes by respectively tagging 3 oligonucleotides x', y', z' with fluorescent, fixation of untagged DNA strands to the surface by means of a connection of 6 to 9 atoms where the DNA strands are arranged in 3lines and 8 rows representing all variables of the given computational problem (see Figure 1).

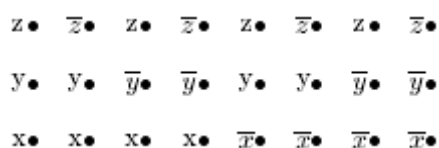


Figure 1. Fixed untagged DNA strands on the surface.

For the first constraint equation, we add the complementary strands x', y' and z' corresponding to variable x, y, z to the surface. Any solution satisfied this inequality will be hybridized at least 2 complementary strand tagged with a fluorescent label (at least 2 bright point), Further, we can determine the solution that satisfy constraint equation by a method of fluorescence-image, and observe their color and record (the feasible solution of the problem is “3,5,6,7”, see Figure 2).

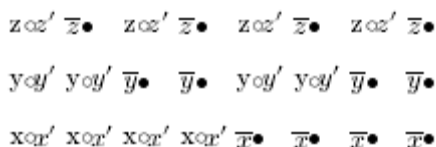


Figure 2. Hybridize figure of the first constraint equation.

(3) The temperature is raised to separate all double-stranded DNA into single-strands by thermal denaturation. The surface is returned to the initial state by washing in buffer (without regard for infeasible solution determined above).

(4) For second constraint equation, similar to step (2), (3) above by adding the complementary strands x', z' corresponding to variable x, z to the surface, we can determine the solution of satisfying constraint equation by a method of fluorescence-image, and observe their color and record, Any solution satisfied this inequality will be hybridized at most 1 complementary strand tagged with a fluorescent (the feasible solution of the problem is “1,3,4,6”, see Figure 3), repeat step (2), (3) above.

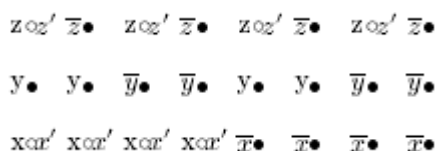


Figure 3. Hybridize figure of the second constraint equation.

(5) For third constraint equation, similar to step (2), (3) above by adding the complementary strands x', y' corresponding to variable x, y to the surface, any solution satisfied this inequality will be hybridized at least 1 complementary strand tagged with a fluorescent green (the feasible solution of the problem is “2,3,4, 5,6,7”, see Figure 4).

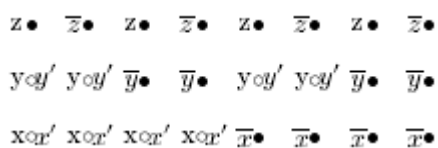


Figure 4. Hybridize figure of the third constraint equation.

(6) There are two feasible solutions “3,6” in the problem, corresponding to variable value is (1,1,0), (0,1,1). By comparing to the value of the object function corresponding to every feasible solution, we can obtain optimum solution (0,1,1), and the minimum value of object function is 4.

The experiment is not complicated and we can accomplish a result that is similar to the experiment performed by Wu [8].

4. CONCLUSIONS

Because computers have obvious limits in storage, speed, intelligence and miniaturization, recently, concerns regarding the methods of DNA computation have arisen, especially their efficient parallelism. In order to solve a practical issue, there are still some problems that need a farther study in biologic technology. In this article, we highlight a DNA computing model to solve a problem of the simple of 0–1 programming problem.

The model we proposed has a potential to solve linear programming problem, which is an important issue in operational research. With the advance of the biologic technology and the molecule biology, the general linear programming problem will be solved. In our method, we adopt fluorescence marking technique and laser focus technique, and read solution by viewing fluorescence, the method of which has some significant advantages such as low cost, low error, short operating time, reusable surface and simple experimental steps.

Acknowledgment

The authors sincerely appreciate the encouraging comments of the Editor of the journal on this paper. They also wish to thank an anonymous referee of this paper who provided many useful and constructive suggestions for the improvement of this paper. Finally, thanks to all the authors that appear in the references. This project is supported by the National Natural Science Foundation of China (Grant No. 60274026) and Anhui province educational committee Foundation (Grant No. Kj098).

5 REFERENCES

- [1] R. P. Feynman, *In minaturization*; Gilbert, D. H., Ed.; Reinhold: New York, **1961**, 282–296.
- [2] Q. Ouyang, P. D. Kaplan, S. M. Liu, A. Libchaber, DNA Solution of the Maximal Clique Problem. *Science* **1997**, 278, 446–449.
- [3] H. Christos, Papadimitriou, *Combinatorial Optimization: Algorithms and Complexity*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey **1989**.
- [4] J. C. Cox, The Complexities of DNA Computation. *TIBTECH* **1996**, 17, 151–154.
- [5] L. M. Adleman, Molecular Computation of Solutions to Combinatorial Problems. *Science* **1994**, 266, 1021–1023.
- [6] R. J. Lipton, DNA Solution of Hard Computational Problems. *Science* **1995**, 268, 542–545.
- [7] Q. Liu, *et al.* DNA Computing on Surfaces, *Nature* **2000**, 403, 175–179.
- [8] H. Y. Wu, An Improved Surface–based Method for DNA Computation, *Biosystem* **2001**, 59, 1–5.
- [9] Z. X. Yin, *et al.* A Chinese Postman Problem Based on DNA Computing, *J. Chem. Inf. Comput. Sci.* **2002**, 42, 222–224.

- [10] T. Head, Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bull. Math. Biol.* **1997**, 49, 737-759.
- [11] Q. Liu, *et al.* Progress toward demonstration of a surface based DNA Computation: a one word approach to solve a model satisfiability problem, *Biosystems* **1999**, 52, 25-33.
- [12] K. Sakamoto, G. Hidetaka; *et al.* Molecular Computation by DNA Hairpin Formation. *Science* **2000**, 288, 1223-1226.
- [13] R. S. Braich, Solution of a 20-Variable 3-SAT Problem on a DNA Computer. *Science* **2002**, 296, 499-502.
- [14] J. A. Bondy, U. S. R. Murty, *Graph Theory with Applications*, The Macmillan Press LTD. Newyork **1976**.
- [15] S.L. Gass, *Linear programming Methods and Applications*, Fifth Edition, Mc Graw Hill Book company **1988**.

Biographies

Yin-zhi Xiang is professor of mathematics at the Anhui University of Science and Technology. After obtaining a Ph.D. degree in mathematical modeling of DNA computing from the Huazhong University of Science and Technology, Dr. Yin-zhi Xiang undertook DNA computing research at the Anhui University of Science and Technology.