

# A DNA Computing Model to Solve 0-1 Integer Programming Problem

**Sanchita Paul**

Lecturer, Department of Computer Science & Engineering  
Birla Institute of Technology, Mesra, Ranchi Jharkhand-835215, India  
sanchita07@gmail.com

**G. Sahoo**

Head of Department, Department of Computer Science & Engineering  
Birla Institute of Technology, Mesra, Ranchi Jharkhand-835215, India  
gsahoo@bitmesra.ac.in

## Abstract

The topic of binary optimization in integer linear programming is an intensive research area in the field of DNA computing. In this paper, a new DNA computation model utilizing solution-based methods and fluorescence labeling techniques has been discussed for getting the optimal solution of the 0–1 integer programming problem. This model describes the major benefits of solution-based methods including vast parallelism, extraordinary information density and ease of operation. In addition, the merit of this model is its automation characteristics, simple coding and time efficiency.

**Keyword:** DNA computing, fluorescence material.

## Introduction

The field of DNA computing is concerned with possibility of performing computations using biological molecules. It provides an understanding how complex biological molecules process information in an attempt to gain insight into new models of computing. DNA computing is interested in applying computer science methods and models to understand such biological phenomena and gain interest into early molecular evolution and origin of biological information processing. Though

the speed of computing becomes faster and the capacity of memory becomes more immense, many complex problems (e.g. NP-complete problems, prediction of protein structure and function etc.) cannot be solved by electronic computer. Many novel methods for the computational purpose have been discussed in the literature in order to settle these kinds of problems efficiently. DNA Computing is a method designed for solving a class of intractable problem, in which the computing speeds up exponentially with the problem size. Using DNA molecules for computing is based on its high density of information storage and computational capabilities.

The DNA (deoxyribonucleic acid) is a double helix with two coiled strands (chains); each composed of nucleotide units and one strand is complementary to the other. Every nucleotide consists of phosphate, sugar and one of the following bases: adenine (abbreviated as A), thymine (T), guanine (G) and cytosine (C). The two chains are held together by hydrogen bonds, which exist only between pairs of complementary bases such as, intermolecular, A & T and G & C. Due to this favored intermolecular interactions particular molecules recognize each other and as a result of which a kind of key-lock decoding information is possible.

In addition to classical methods meant for solving integer-programming problem, some molecular computing models are discussed in the literature and are based on the primary trends in research studies known as solution based and surface based DNA computations.

Surface-based DNA computation, which manipulates DNA strands immobilized on a surface [1, 2, 3], is another significant progress in DNA computation. This method normally uses more advanced biological technology and has the potential to be automated in contrast to the solution-based method. However, the inherent disadvantage of surface based models is that the number of DNA strands involved in computation is limited since the strands are restricted to two dimensions of a surface rather than three (Yin et al., 2007 [4]). When the number of DNA strands increase with problem size, it is impossible to solve in surface base method. In this paper, we proposed a new DNA computing model based upon solution-based method to solve 0–1 programming problem. Our approach involves the incubation of solution-space that contains the “sticker” DNA strands. Adleman and co-workers (1998) [7] introduced an abstract model of molecular computation called the “sticker model”, which has random access memory to exploit information encoding. The sticker model breaks the strand into bit of strings where each bit comprises several DNA molecules. A sticker is complimentary to one and only one bit string.

On the basis Adleman’s experiment we present a solution base model to solve 0-1 integer programming problem. The rest of the paper is organized as follows: Section 2 introduces the 0-1 integer programming problem; section 3 describes the DNA algorithm for 0-1 integer programming problem; section 4 describes an example of the DNA algorithm for solving 0-1 integer programming problem following by conclusion.

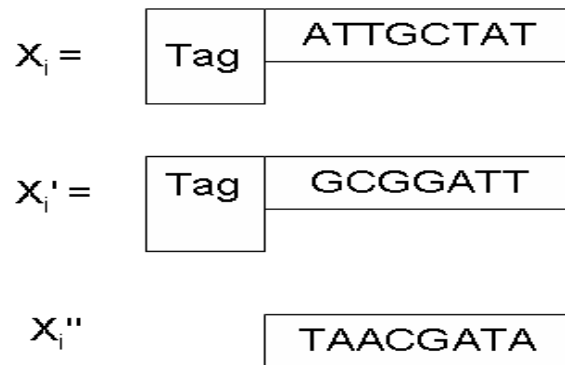
## **2. The 0-1 programming problem**

The 0-1 programming problem is a special case of an integer-programming problem,



### 3. Model Description and DNA algorithm for binary integer (0-1) problem:

This model involves a system of equations that contains  $n$  variables  $x_1, x_2, \dots, x_n$ , and  $m$  equations. Each variable is represented by a single stranded DNA stretch with a double stranded tag at the beginning. This imparts a sticky end to each variable as shown in the fig 2. Similar structures with different composition of nucleotides and tags are to be taken which denote the false values,  $x_1', x_2', \dots, x_n'$ . The constraints are provide to the solution space by using  $x_1'', x_2'', \dots, x_n''$  strands which are complementary to the single stranded portion of the variables  $x_1, x_2, \dots, x_n$ . These  $x_1'', x_2'', \dots, x_n''$  attach to their respective complementary portions on the variables which are  $x_1, x_2, \dots, x_n$ . With the help of fluorescent tagged material we can read out our required solution. By using  $x_1'', x_2'', \dots, x_n''$  we provide all the given constrains in different pools where each pool satisfy one of the given constraints and screen out our solution space to a list of feasible solutions. Then every value of objective function is compared to every feasible solution to get an optimal solution. For more clarity we discuss in detail about the model for solving 0-1 integer programming in such a manner:



**Fig 2: Representation of variables using DNA strands**

The flow chart of proposed DNA algorithm for solving 0-1 integer programming problem is summarized as follows:

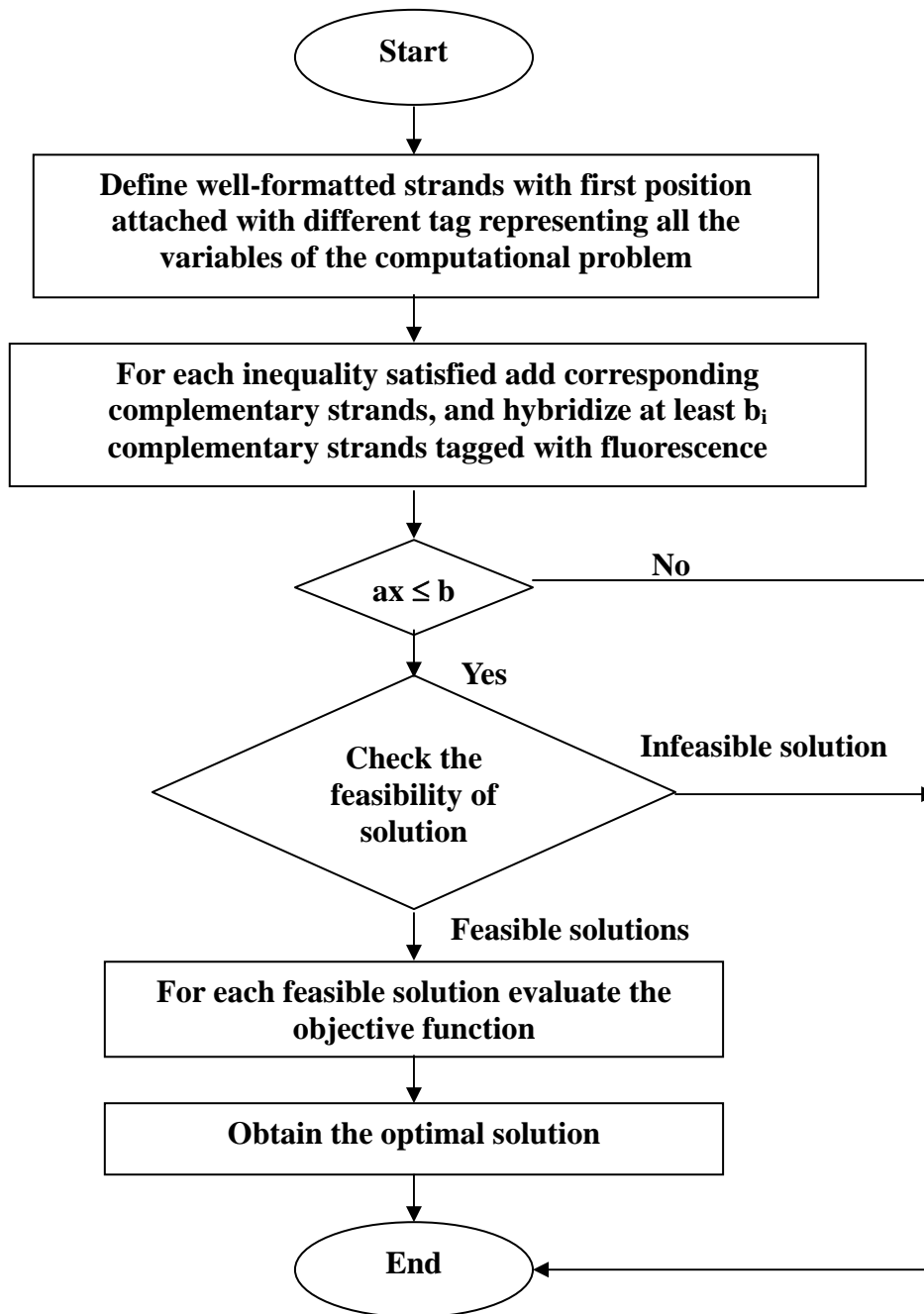


Fig3: DNA Algorithm for solving 0-1 integer programming problem

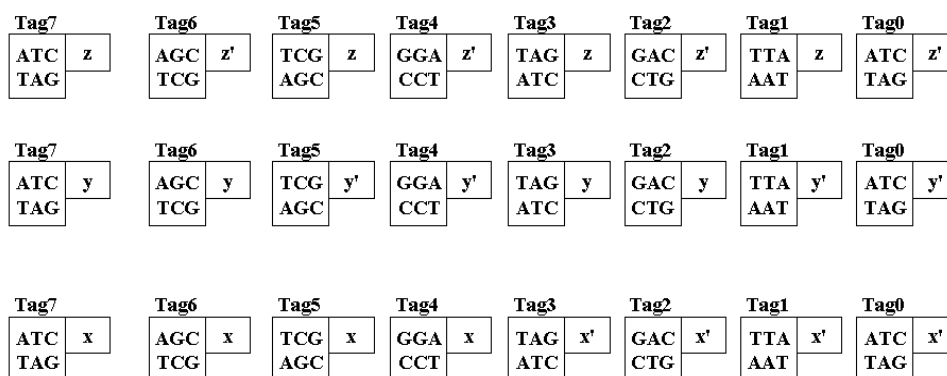
#### 4. An example of 0-1 integer programming problem

$$\begin{array}{l}
 \text{Min } u = 2x + 3y + z \\
 \text{Subject to the constraints} \\
 \left\{ \begin{array}{l}
 x + y + z \geq 2 \\
 x + y \geq 1 \\
 y + z \leq 1 \\
 \text{Where, } x, y, z = 0, 1
 \end{array} \right.
 \end{array}$$

The process for solving 0-1 programming problem is divided into following six steps:

**Step 1:** In a container we first synthesized 9 oligonucleotides, which were divided into 3 groups. The oligonucleotides of the first group represented variables  $x, y, z$  attached with different tags; the oligonucleotides in the second group similarly represented variables  $x', y', z'$ , also attached with different tags ( $x=1$  if and only if  $x'=0$ , such as  $y, z$ ); the oligonucleotides in third group represented the complementary strands of the first group (without any tags) and are denoted as  $x'', y'', z''$ .

**Step 2:** We generate different combination of DNA molecules where we choose oligonucleotides  $x, y, z$  and  $x', y', z'$  such that they must be very different, oligonucleotide  $x$  represents variable  $x = 1$  and oligonucleotide  $x'=0$ , same for  $y, z$ . (see fig 4).



**Fig 4:** Combination of oligonucleotides are placed in a container

**Step 3:** Copies of the first container's molecules are placed in 3 different containers for 3 equations. Total process is done in parallel and takes less time.

**Step 4:** According to first equation, we added DNA probes, respectively tagging 3 oligonucleotides  $x'', y'', z''$  with fluorescent material (chemical compound and green in color). For first constraint equation, we passed the complementary strands  $x'', y'', z''$  tagged with fluorescent material corresponding to variable  $x, y, z$ . Any solution satisfying this inequality is hybridized with at least 2 complementary strands tagged with a fluorescent material (at least two bright point) and the feasible solution of the problem is "7,6,5,3". (See fig 5. where under tag7- 3 bright points are visible, under tag6- 2 bright points are visible, under tag5- 2 bright points are visible, and under tag 3- 2 bright points are visible which satisfies the first equation).

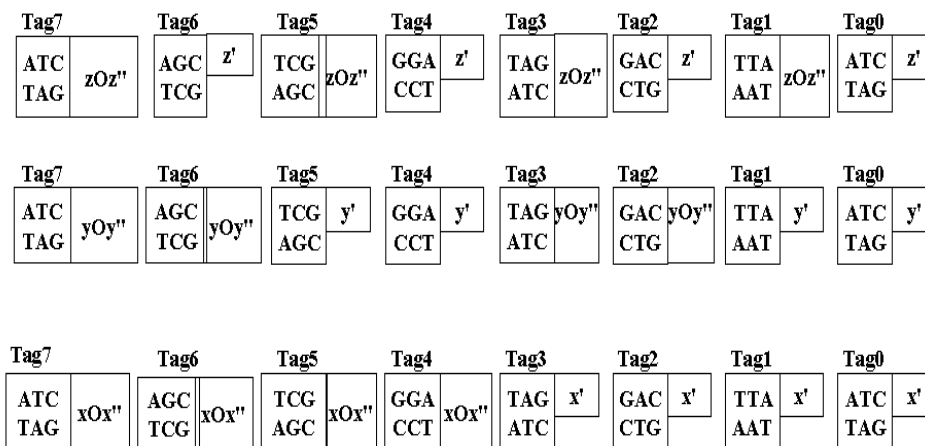


Fig5: Hybridized fig of the first constraint equation.

**Step 5:** Steps 3 and 4 is repeated for second constraint equation, by adding the complementary strands  $x''$ ,  $y''$  corresponding to variable  $x$ ,  $y$ . Any solution satisfying this inequality is hybridized with at least 1 complementary strand tagged with a fluorescent material (the feasible solution of the problem is “7, 6, 5, 4, 3, 2”, see fig6).

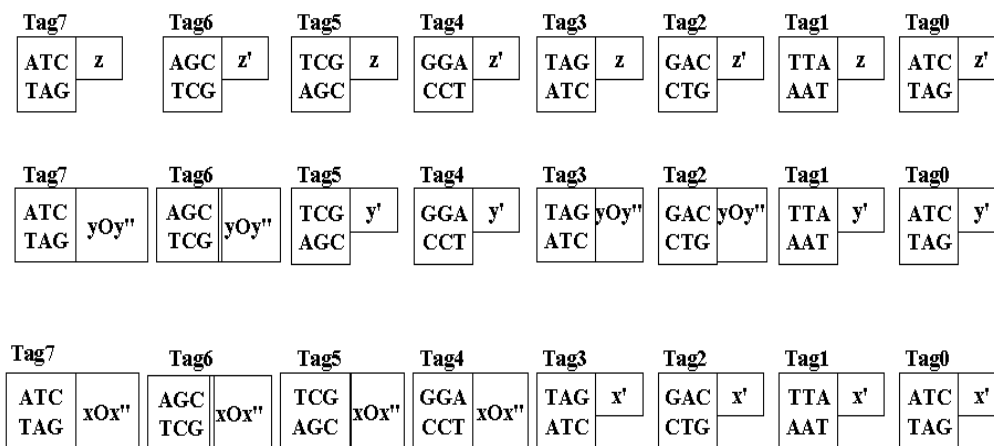
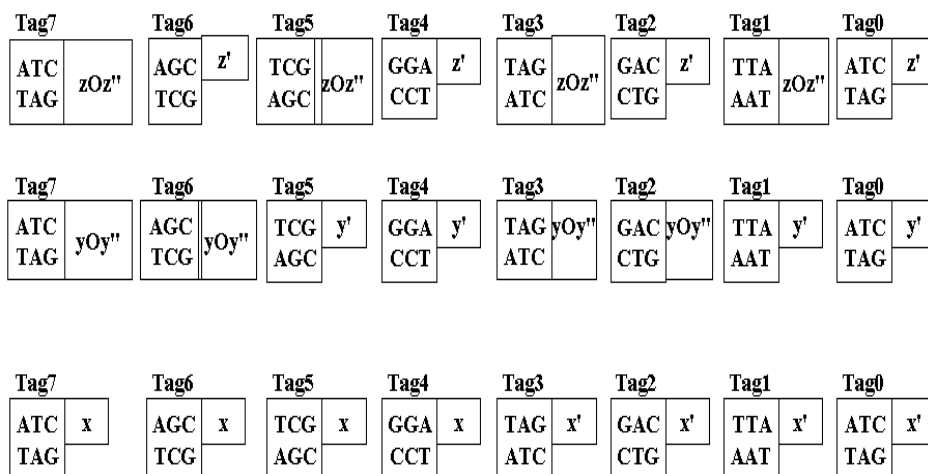


Fig 6: Hybridized fig of the second constraint equation.

**Steps 7:** Step 3 and 4 is repeated for third constraint equation, by adding the complementary strands  $y''$ ,  $z''$  corresponding to variable  $z$ ,  $y$ . Any solution satisfying this inequality is hybridized with at most 1 complementary strand tagged with a florescent material (the feasible solution of the problem is “6, 5, 4,2,1,0” see fig 7).



**Fig 7: Hybridized fig of the third constraint equation.**

**Result:** There are finally two feasible solutions “6,5 ” in the problem, corresponding to variable (1,0,1) and (0,1,1.) By comparing to the value of object function corresponding to every feasible solution, we can obtain optimum solution (1, 0, 1) and minimum value of object function is 3.

**Discussion:** Electronic computer obviously has limits in storage, intelligence and miniaturization. Recently concerns regarding the methods of DNA computation have arisen, especially for its parallelism. In this article we proposed a DNA computing model in detail to solve a simple 0-1 programming problem. The advantage of this model is when the size of the equation increases the number of DNA strands is also increased., in contrast to surface-base model where there is some limitation about space and the strands are restricted to two dimensions of a surface rather than three (Yin et al., 2007). But this model has no such type of restrictions for solving 0-1 programming problem.

In this method the use of fluorescent material for reading the feasible solution is advantageous since it involves low cost, low error, and simple experimental steps.

**Conclusions:** This model encompasses the major benefits of solution-based methods; including vast parallelism, extraordinary information density and ease of operation. The other significant advantages of our method are low error, short operating time, less manual labor, reusable model with simple experimental steps.



## References

- [1] E. Anne, W. Cai, M Robert et al., The power of surface-based DNA computation, In: Proceedings of the First Annual International Conference on Computational Molecular Biology, 1996, 67–74.
- [2] G. Frutos, M.Corn, M.Smith et al., Enzymatic ligation reactions of DNA “word” on surface for DNA computing. *J.Am. Chem. Soc* 120(40), 1998, 10277–10282.
- [3] A.G.Fruto, L.M. Wang, Q.H. Liu, et al., DNA computing on surfaces. *Nature* 403 (13), 2000, 175–179.
- [4] F.Y.Zhang, J.Xu, Z.X.Yin, The general from of 0-1 programming problem based on DNA computing, *Biosystems* 70 (2003) 73-78.
- [5] L.M. Adleman, Molecular computation of solution to combinatorial problem, *Science* 226 (1994) 1021-1024.
- [6] H.Y.Wu, An improved surface based method for DNA computation, *Bio-systems* 59 (2001) 1-5.
- [7] L.M. Adleman, W.K.Rothmund Paul., et al., A sticker based model of DNA Computation, *Biosystems* 5(4) (1998) 615-629.
- [8] Y.Sun, Z. Wang, The genetic algorithm for 0-1 programming with linear constraints, In: D.B.Fogel (Ed), *Proceeding of the 1<sup>st</sup> ICE’ 94*, Orlando, FL, 1994, 559-564.

**Received: Aptil 1, 2008**