Solution of 0-1 Programming Problem by using DNA Computing Model

Navin Kumar Agrawal¹, Rajeev Kumar² and Rahul Rathore³

¹Department of Computer Science, Teerthanker Mahaveer University, Moradabad, (U.P.)
²Department of Computer Application, College of Management & Computer Application, Teerthanker Mahaveer University Moradabad (U.P.)
³Department of Computer Science, Teerthanker Mahaveer University, Moradabad, (U.P.)

Email: rajeev2009mca@gmail.com, garg.gla@gmail.com

Abstract: In this problem, DNA computing model is used with fluorescence labeling techniques to solve the 0-1 integer problem. This model was motivated by parallelism and extra ordinary information density. The merit of this model is ease of Operation, simple implementation and time efficiency. The use of fluorescent material for reading the feasible solution is advantageous since it involves low cost, low error, and simple experimental step.

Introduction:
The field of DNA computing is use to perform the computation with the DNA molecule.DNA molecule is a nucleic acid that contain the genetic information of instruction that are used the development and functioning of cellular organism.

0-1 integer programming problem is a linear integer programming problem where all variables are restricted to take values of either 0 or 1. This problem is NP-hard, and as such, it is considered unlikely that there exists an efficient algorithm for solving it.

The main role of DNA molecule is the long-term storage of information [6].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. The DNA molecules is use for representing information, and also the first attempt to deal with an NP-complete problem while NP-complete problem cannot be solved by electronic computer. 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.DNA consist of polymer chain, also called DNA strands. Each strand may be viewed as a chain of nucleotides, or bases. 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[4,5,6].

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 [6].

2. The 0-1 Programming problem:
The 0-1 programming problem is a special case of an integer-programming problem, in which the value of every variable is restricted to be 0 or 1. Its general form can be defined as:

\[ z(\text{max or Min}) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \]

Subject to

\[ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \]
\[ \vdots \]
\[ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \]

Where, \( x_i \) = 0 or 1, \( 1 \leq i \leq n \) and \( b_j \) are non-negative integers, \( 1 \leq j \leq m \). We present a model to solve the general 0-1 programming problem with DNA, when \( a_{ij} \) is an integer value. Since \( a_{ij} \) is an integer, each constraint subjected to optimization function can be

\[ a_{ij}x_1 + \ldots + a_{ijn}x_n \leq b_i \]

\[ a_{ij}x_1 + \ldots + a_{ijn}x_n \geq b_i \]
transformed into corresponding constraints where $a_{ij}$ takes the value as 0 and 1.

The flow-chart corresponding to the overall scheme of algorithm for solving 0-1 programming problem is depicted in fig 1.

Fig 1: Overall flow of algorithm

3. DNA algorithm for binary integer (0-1) problem:
This problem involves a set of equations where \( m \) is the number of equations and each equation contains \( n \) variables \( x_1, x_2, ..., x_n \). Each variable is represented by a single stranded DNA stretch with a double stranded tag at the beginning \([6]\). 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 \( x_1', x_2', x_3' \ldots \ldots \ldots, x_n' \). The constraints are provide to the solution space by using \( x_1'', x_2'', ..., x_n'' \) strands which are complementary to the single stranded portion of the variables \( x_1, x_2, ..., x_n \). These \( x_1'', x_2'', ..., x_n'' \) attach to their respective complementary portions on the variables which are \( x_1, x_2, ..., x_n \). With the help of fluorescent tagged material we can read out our required solution. By using \( x_1'', x_2'', ..., x_n'' \) 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:

\[
\begin{align*}
X_i &= \text{ATTGCTAT} \\
X_i' &= \text{GCGGATT} \\
X_i'' &= \text{TAACGATA}
\end{align*}
\]

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:

Start

Define strands with first position attached with different tags representing all variable

For each inequality satisfied add corresponding Complementary strands, and hybridize at least bi Complementary strands tagged with fluorescence

No

\( ax \leq b \)
Fig 3: DNA Algorithm for solving 0-1 integer programming problem

Discussion:
Electronic computer obviously has limits in storage and intelligence. In this paper, we proposed a DNA computing model to solve a simple 0-1 programming problem by using methods of DNA computation. 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 includes the major benefits of solution-based methods like 0-1 programming problem 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.

References:


