Creative Conceptual Design Based on Evolutionary DNA Computing Technique

Xiyu Liu[1], Hong Liu[2], Yangyang Zheng[1]

[1]School of Management and Economics, Shandong Normal University
Jinan, 250014, P. R. China
[2]School of Information Science and Engineering, Shandong Normal University
Jinan, 250014, P. R. China
{Xiyu.Liu, Hong.Liu,Yangyang.Zheng, sdxyliu}@163.com

Abstract. Creative conceptual design is an important area in computer aided innovation. Typical design methodology includes exploration and optimization by evolutionary techniques such as EC and swarm intelligence. Although there are many proposed algorithms and applications for creative design by these techniques, the computing models are implemented mostly by traditional von Neumann's architecture. On the other hand, the possibility of using DNA as a computing technique arouses wide interests in recent years with huge built-in parallel computing nature and ability to solve NP complete problems. This new computing technique is performed by biological operations on DNA molecules rather than chips. The purpose of this paper is to propose a simulated evolutionary DNA computing model and integrate DNA computing with creative conceptual design. The proposed technique will apply for large scale, high parallel design problems potentially.

Keywords: Evolution, conceptual design, DNA computing, innovation.

1 Introduction

Deoxyribonucleic acid computing, or DNA computing in short, has attracted strong interests and wide focuses recently. DNA computing is in a sense similar to evolutionary computing but the significant difference between them lies in the computing medium: biomolecules rather than transistor chips. The essential work to reveal the ability of DNA in computing is by Adleman's experiment (Adleman [1]). His work was later generalized by Lipton [12] to the satisfiability problem. Based on these, a number of applications in solving problems such as factorization, graph theory, control and nanostructures have emerged, as well as theoretical studies including DNA computers, see [15][5][10][11][17][6][7][14] for references. However, the effective combination of evolutionary computation technique with DNA computing is rarely seen due to the difficulty in checking a good molecule as solution to the computing problem with the relatively restricted biological enzyme operations.

On the other hand, although the concept of design has changed significantly, the works that designers do has not changed much when they are involved in creative design [3]. Frequently, design is called art because it is started with ideas from magazines, journals and photographs of similar or rival designs. Designers turn these concepts into detailed, concrete, clearly defined constraints, parts and design pieces during which they are tested, evaluated, and finally accomplished to design works.

Up to now the combination of DNA computing and cluster analysis can be only found in one research; see Rohani Binti Abu Bakar et al [14]. Inspired by the previous research, this paper focuses on the joint study of DNA computing with creative conceptual. We propose an evolutionary model for DNA computing. We also presents two encoding scheme for the design problem. An experimental framework is also given.

2 Strands and Standard Operations

2.1 DNA Structures

In DNA, the nucleotides are the purines adenine (A), guanine (G), the pyrimidines thymine (T) and cytosine (C). The single strands of DNA can form a double-stranded molecule when the nucleotides hydrogen bond to their Watson-Crick complements, $\overline{A} = T$ and $\overline{G} = C$ (Figure 1).

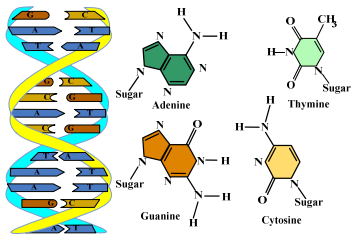


Fig. 1. A double-stranded DNA structure diagram.

DNA stores information in nucleic acid and manipulates information via enzymes and interactions. Double strand in Fig 2 presents a DNA molecule composed of ten pairs of nucleotides. Bonding occurs by the pair wise attraction of bases. The pairs (A, T) and (G, C) are called complementary base pairs.

There is another structure of DNA molecules called the hairpin structure. The DNA forms a partial double strand in hairpin formation [10]. Figure 3 is a hairpin example formed from the next long example strand.

$$5'—ACTGTTAAGAGGGGATAGTGTATTCTTAACAGT —3' \quad (1)$$

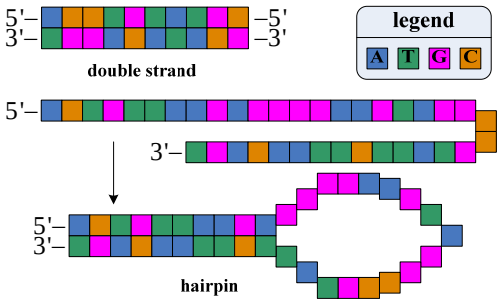


Fig. 2. A diagram of nucleotides sequences in DNA strands and a single strand DNA forms a hairpin formation.

The cutting of certain strands is performed by restriction enzymes at certain recognition sites. Recognition sites are typically 4-8 DNA base pairs long. Figure 3 shows some examples.

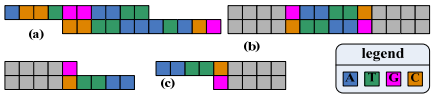


Fig. 3. DNA molecules with sticky ends.

There are over 100 different restriction enzymes, each of which cuts at its specific recognition site. A restriction enzyme cuts DNA into pieces with sticky ends. On the other hand, sticky ends will match and attach to other sticky ends of any other DNA that has been cut with the same enzyme. DNA ligase joins the matching sticky ends of the DNA pieces from different sources that have been cut by the same restriction enzyme.

2.2 Coding and Operations for DNA

The basic operations performed by enzymes are denaturing, replicating, merging, detecting etc. Some of these operations are similar to simulated genetic operations. In order to simulate DNA computing without tubes, we need to code the DNA strands and operations into formal language symbols. Notice that the pairs (A, T) and (G, C) are complementary base pairs. We can easily design a coding scheme for the four bases: $A=00, G=01, C=10, T=11$.

Biologically, the basic DNA operations available on DNA are mainly:

- Merge $m(N_1, N_2) \Box N_1 \cup N_2 = N.$
- Amplify *duplicate*(*N1*) = N.
- Detect(N).

- Separate or extract. Given a word *w* consisting of strings from $\sum=\{A,G,C,T\}$ and a tube N, generate two tubes $+(N,w)$ and $-(N,w)$ which contains and does not contains the string w: $N \leftarrow +(N,w), N \leftarrow -(N,w)$.
- Length separate. Given a tube *N* and an integer *n*, generate a tube containing strands with length less or equal to *n*: $N \leftarrow (N,\leq n)$.
- Position separate. Given a tube and word generate a tube with stands beginning(ending) with the word: $N \leftarrow B(N_1,w); N \leftarrow E(N_1,w)$.

Hybridization is reassociation techniques of single-stranded DNA to form double-stranded DNA when one of the strands originates from one organism and the other strand from another organism. Hybridization occurs when the base sequences are complementary or nearly so.

There are several types of hybridizations. The first one is the complete hybridization which takes place between two complementary DNA single strands strictly by the Watson-Crick rule. The second is the false positive hybrid. The third is the false negative hybrid between completely complementary strands and they are not completely hybridized due to some errors. The hairpin hybrid is another type of hybridization.

Hybridization is a main process in DNA computing to form all possibilities of solution strands in which the right answer lies.

3 Formal DNA Computational Models

There are three types of DNA computing techniques: intramolecular, intermolecul-ar, and supramolecular. Intermolecular DNA computing focuses on the hybridization between different DNA molecules as a basic step of computations. Since the original work of Adelman and Lipton, nearly all the current DNA computing strategies are based on enumerating all candidate solutions, and then using some selection process to choose the correct DNA. This technique requires that the size of the initial data pool increases exponentially with the number of variables in the calculation. As the problem size keeps increasing, this brute-force searching method will become intolerable. Therefore the combination of artificial intelligence in DNA computing is necessary to improve its effciency. In order to achieve this goal, a kind of formal description is essential.

3.1 The Sticker Model

Sticker model is based on a coding scheme called DNA complex. A DNA complex is a partially double DNA strand. Usually a double piece represents a bit with value one while a single strand represent zero. Hence each complex is constructed by two single stranded DNA molecules referred to as memory strands and sticker strands.

The main operations of the sticker model are merging, separating, setting, and clearing which are standard biological operations performed in a test tube. Merging operation just unites two input test tubes and combines them into one test tube as

described in Section 2. The separating operation separates the test tube into two test tubes according to the value in a specified bit. The value of strands at this bit is ”1” in one of test tube, and ”0” in the other tube. Setting operation turns the value of a specific bit for all strands into ”1”. On the other hand, clearing operation turns the value of a bit into ”0”.

The sticker model, or the sticker system, is a formal language generative device based on the sticking operations which is an abstract model of sticking operation used in DNA computing.

3.2 The Splicing Model

Tom Head [7] proposed a splicing model based on formal language theory. A splicing system $S = (A, L, B, C)$ consists of a finite alphabet A, a finite set I of initial strings in $A*$(language over A) , and finite sets B and C of triples (c, x, d) with $c, d, x \in A*$.Each such triple in B or C is called a pattern. For each such triple the string cxd is called a site and the string x is called a crossing. Patterns in B are called left patterns and patterns in C are called right patterns. The language $L = L(S)$ generated by S consists of the strings in I and all strings that can be obtained by adjoining to *Lucxfq* and *pexdv* whenever *ucsdv* and *pexfq* are in L and (c, x, d) and (e, x, f) are patterns of the same hand. A language L is a splicing language if there exists a splicing system S for which $L = L(S)$.

In this way, the author proposed a formal language description to the study of DNA computing. A language is associated with each pair of sets where the first set consists of double stranded DNA molecules and the second set consists of the recombinational behaviors allowed by specified classes of enzymatic activities. The associated language consists of strings of symbols that represent the primary structures of the DNA molecules that may potentially arise from the original set of DNA molecules under the given enzymatic activities.

3.3 The k-armed Model

The k-armed model is based on some more complicated molecule structures which have three-dimensional DNA architecture as shown in Figure 4([9]). Although the armed molecules are relatively less well known, they have been found existed in real molecules and researches on the construction and stability have begun.

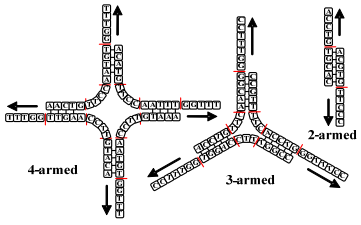


Fig. 4. k-armed DNA architecture.

It is natural to use the armed model to represent SAT problem in terms of contact network framework and give theoretical solutions to this NP complete problems [9]. Due to the natural similarity to clusters, the k-armed model is good candidate in the description of cluster analysis.

The cycle model is an artificial model. In this model, the DNA sequence forms a double stranded cycle as shown in Fig. 5. There are two another arms in this model with sticky ends. Molecules can join into chains via these arms for form a chain model.

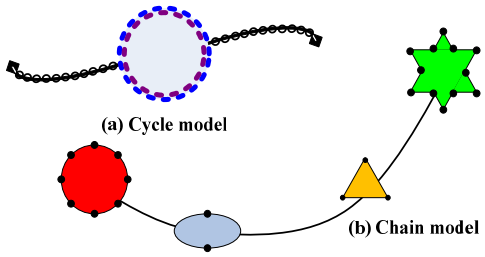


Fig. 5. Cycle and chain DNA architecture.

4 Evolutionary DNA Computing Model

In this section, we will propose an evolutionary variant splicing model based on the work of Tom Head [7]. In order to simplify the symbols, we will use *a,c,g,t* to represent the four deoxyribonucleotides. Then the alphabet is defined by

$$\sum = \{A, C, G, T\}, A = \left|\begin{matrix} a \\ t \end{matrix}\right|, T = \left|\begin{matrix} t \\ a \end{matrix}\right|, G = \left|\begin{matrix} g \\ c \end{matrix}\right|, C = \left|\begin{matrix} c \\ g \end{matrix}\right| \tag{2}$$

Let $\sum *$ be the set of all finite strings consisting of symbols from the alphabet. Then a member of $\sum *$ is a double stranded DNA which will be denoted by lower case Greek letters later. The natural involution operator on $\sum *$ is defined by

$$\begin{cases} f(A) = T,\ f(G) = C \\ f(f(\alpha)) = \alpha,\ f(\alpha\beta) = f(\alpha) f(\beta) \end{cases} \tag{3}$$

A subset $P \subset \sum *$ is called a population. We use the symbol S_{ez} to represent a restriction enzyme in the form of $\pm(\alpha, \beta, \gamma)$ where α and γ are the cutting sites and β is the cleavage sequence. The symbol $\pm$ indicates that the left cutting is at the top

strand $(5'—)$ end, or the bottom strand $(3'—)$ end. In Table 1 there are some examples of restriction enzymes. The set of all possible restriction enzymes is $S = \{S_{ez}\}$.

Table 1. Examples of restriction enzymes

Enzyme	Representation	Cutting example
EcoRI	+(G, AATT, C)	xxxxxg aattcxxxxx xxxxxcttaa gxxxxx
TaqI	+(T, CG, A)	xxxxxt ggaxxxxx xxxxxagc txxxxx
S ciNI	+(G, CG, C)	xxxxxg cgcxxxxx xxxxxcgc gxxxxx
HhaI	-(G, CG, C)	xxxxxgcg cxxxxx xxxxxc gcgxxxxx

4.1 Crossover and Mutation

Crossover operation is performed by a restriction enzyme. For one-point crossover, let $\alpha, \beta \in P$ be two strands. Each of them is cut into two strands with sticky ends by certain restriction enzymes, $\alpha = \alpha_L \oplus \alpha_R$, $\beta = \beta_L \oplus \beta_R$. Suppose the sticky ends of α_L and β_R are matching, then the crossover operation will result two new strands $\alpha = \alpha_L \oplus \beta_R$ and $\beta = \beta_L \oplus \alpha_R$.

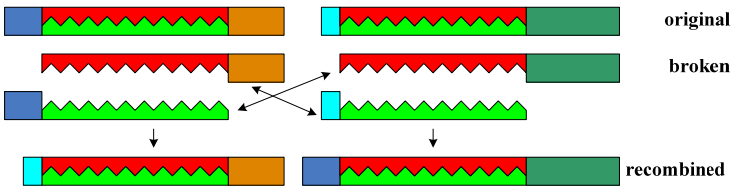


Fig. 6. Crossover of DNA molecules.

Crossover can be performed by controlled hybridization. First we use all kinds of restriction enzymes to break the double strands. Then by ligation DNA sequences recombine with each other in a test tube by means of enzyme reaction. At the end of this process, new DNA sequence will ligate together to form a new DNA strings. Figure 10 shows the process of crossover operation.

Mutation will randomly select one or several successive bases and replace them with other bases. For example, we can choose a base A and change it into G. The operation of mutation is carried out by insertion and deletion technique. First we can locate the DNA subsequence and delete sequence by melting and annealing. Then we can insert a new sequence by PCR and ligation operations.

4.2 Selection and Fitness Computation

Before selection, we need a procedure similar to the sort operation as in simulated genetic computing. This is based on the coding, that is, we encode the DNA strands corresponding to the fitness of solutions. Good DNA strands have larger length while less fitted DNAs have smaller length as shown below.

$$\begin{vmatrix} \text{problem encoding} & \text{fitness encoding} \\ \text{equal-width} & \text{length} \propto \text{fitness value} \end{vmatrix} \tag{4}$$

Then the elimination of less fitted DNA can be performed by gel electrophoresis operations. The step to add new DNA sequences can be implemented by polymerase chain reaction (PCR). PCR is a process that quickly amplifies the amount of specific molecules of DNA in a given solution using primer extension by polymerase. DNA polymerases perform several functions including the repair and duplication of DNA. By PCR reaction we can double the quantity of specific DNA molecules.

From the above discussion, we propose the DNA evolution procedure as in Table 2.

Table 2. DNA Evolution Procedure

1. Generate suitable amount of DNA sequences and place them in a test tube.
2. Generate enough DNA complements and mutation sequences as desired. Prepare desired enzymes.
3. Perform crossover and mutation operations.
4. Perform selection and fitness computation..
5. Eliminate a number of worst DNAs and substitute with new DNA sequences.
6. Check best one by DNA length to determine if it is the desired solution.
7. Loop.
8. The final obtained DNA sequence shows the solution

5 DNA Based Design Exploration

Although the concept of design has changed significantly, the works that designers do has not changed much when they are involved in creative design [3]. Designers obtain ideas and turn these concepts into detailed, concrete design works. Figure 7 gives a rough sketch of the main processes of creative design and how they interact with one another. Finding the most interesting alternative design is then called design optimization.

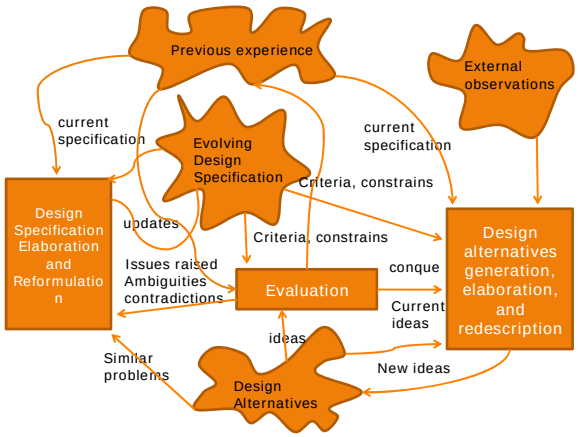


Fig. 7. Creative design as an exploration and optimization (redrawn from Janet et al [8]).

To use DNA computing in design, the first problem is encoding. Along this line there are two kinds of generative systems: the structural and the linguistic [16]. When we focus on the first model, there are two main types of concrete representations: boundary representations and the solid representation. There are three types of solid representation: sweeping, constructive solid geometry, and spatial partitioning [2]. We will use the spatial partitioning in which the atom unit is a curved clipped cube derived from the clipped stretched cube of Peter J. Bentley [2] (Fig. 12). The only difference here is that we use a sphere to replace the cutting plane.

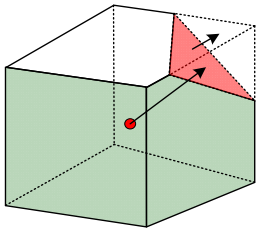
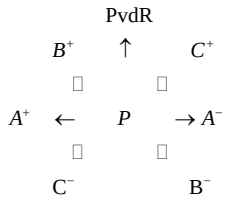


Fig. 8. The Curved Clipped Cube.

Suppose the center of the nonclipped cube, called a primitive, is $P_0 = (x_0, y_0, z_0)$ with width, depth and height *a,b,c*. The distance of the clipping place to the center P_0 is *d*. The unit normal vector of the plane is $v = (\alpha, \beta, \gamma)$ with $\alpha^2 + \beta^2 + \gamma^2 = 1$ .The radius of the sphere is R. We assume that $R > 0$ whenever the center of the sphere is pointed to the center of the cube. Therefore, a curved cube is a dual [*pos,ind*] where *pos* is a vector (P_0, a, b, c, v, d, R) with data dimension 10 indicating the geometrical data of the primitive. The vector $ind = (iA^+, iA^-, iB^+, iB^-, iC^+, iC^-)$ is a six indices vector

indicating the six faces are st-icky or not. A face is called sticky if it can be connected to other face to unit into one object. If a face is cut out all its four vertices, it is not sticky. In all other cases, it can be sticky.

Not we use the 7-armed DNA model to represent a primitive. The model is in for following form



(5)

Here the encoding *P* stands for $P_0 = (x_0, y_0, z_0)$. A^+ consists half of DNA strings of *P* and half of DNA encoding strings of *a*, plus a 1 flag indicating the positive side of the side. A^- is similar to A+ except that the flag is 0. Encodings of B^+, B^-, C^+, C^- are respectively for *b,c*. The string PvdR is encoding of *v,d,R* together with half of *P*. Whenever the indicator $ind = (iA^+, iA^-, iB^+, iB^-, iC^+, iC^-)$ is positive, the corresponding arm is sticky. This allows this arm connective to arms of other primitives.

6 Markov Chain Analysis

Suppose Ω is a finite search space and M is a discrete time stochastic process defined over Ω. Let *P* be the transition matrix with member p_{ij} the transition probability that the next state will be *j* given the current state *i* with the property $0 \le p_{ij} \le 1, \sum_j p_{ij} = 1$ .Since the Markov chain is memoryless [13], the probability p_{ij} depend only on the current state *i*. Let X_t denote the state of the chain at time *t*. Then the memorylessness property is equivalent to the following

$$p_{ij} = p[X_{t+1} = j \,|\, X_t = i] = p[X_{t+1} = j \,|\, X_0 = i_0, ..., |\, X_t = i] \tag{6}$$

For $i, j \in \Omega$, define the t-step transition probability as $p_{ij}^{(t)} = p[X_t = j \,|\, X_0 = i]$. Given an initial state $X_0 = i$, the probability that the first transition into state *j* occurs at time *t* is denoted by $r_{ij}^{(t)}$ and is given by

$$r_{ij}^{(t)} = P[x_t = j, and \text{ for } 1 \le \text{s} \le \text{t-1}, \text{X}_\text{s} \ne j \,|\, X_0 = i] \tag{7}$$

Also, for $X_0 = i$, the probability that there is a visit to (transition into) state j at some time t > 0 is denoted by f_{ij}, and is given by $f_{ij} = \sum_{r>0} r_{ij}^{(t)}$. Finally, the expected number of time steps to reach state j starting from state i is denoted by h_{ij}: $h_{ij} = \sum_{r>0} t r_{ij}^{(t)}$. If $f_{ij} < 1$ then $h_{ij} = \infty$ but the converse need not be true.

Definition 6.1. A state i for which $f_{ii} < 1$ (and hence $h_{ij} = \infty$) is said to be transient, and one for which $f_{ii} = 1$ is said to be persistent. Those persistent states i for which $h_{ii} = \infty$ are said to be null persistent and those for which $h_{ii} \neq \infty$ are said to be nonnull persistent.

We define the underlying directed graph of a Markov chain as follows: there is one vertex in the graph for each state of the Markov chain; and there is an edge directed from vertex i to vertex j if and only if $p_{ij} > 0$.

Definition 6.2. A strong component of a directed graph G is a maximal subgraph C of G such that for any pair of vertices i and j in the vertex set of C, there is a directed path from i to j, as well as a directed path from j to i.

Theorem 6.1 (**Fundamental Theorem of Markov Chains [13]**). Any irreducible, finite, and aperiodic Markov chain has the following properties.

1. All states are ergodic.
2. There is a unique stationary distribution n such that, for $1 \le i \le n, \pi_i > 0$.
3. For $1 \le i \le n$, $f_{ii} = 1, and \quad h_{ii} = 1/\pi_i$.
4. Let N(i, t) be the number of times the Markov chain visits state i in t steps.
5. Then, $\lim_{t \to \infty} N(i,t)/t = \pi_i$

Acknowledgments

This project is carried out under the ”Taishan Scholar” project of Shandong China. Research is also supported by the Natural Science Foundation of China (No.60873058), the Natural Science Foundation of Shandong Province (No. Z2007G03).

References

[1] Adleman L M.: Molecular Computation of Solutions to Combinatorial Problems, Science 266(5187), 1021-1023 (1994).
[2] Bentley, Peter John: Generic Evolutionary Design of Solid Objects using a Genetic Algorithm, Doctoral Thesis, the University of Huddersfild, (1996)
[3] Cho, Sung-Bae: Towards creative evolutionary systems with interactive genetic algorithm, Applied Intelligence 16, 129C138 (2002).
[4] Ding Y.S.: Computational Intelligence: Theory, Technique and Applications (Science Press, Beijing, (2004)
[5] Ezziane Z.: DNA computing: applications and challenges, Nanotechnology 17, R27-R39 (2006)

[6] Hao Yan, Xiaoping Zhang, Zhiyong Shen et al.: A robust DNA mechanical device controlled by hybridization topology, Nature 415(3), 62-65 (2002)
[7] Head Tom: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors, Bulletin of Mathematical Biology 49(6),737-759 (1987)
[8] Janet, L. Kolodner, Linda, M. Wills: Case-Based Creative Design, in: Proceedings Kolodner93 Case-based Creativity, AAAI Spring Symposium on AI and Creativity (Springer, Stanford,CA), pp.50-57. 1993
[9] Jonoska N., Karl S.A., Saito M., Three dimensional DNA structures in computing, Biosystems 52, 143-153 (1999)
[10] Kensaku Sakamoto, et al.: Molecular Computation by DNA Hairpin Formation, Science 288, 1223-1226 (2000)
[11] Ren-Hou Li, Wen Yu: An Exploration of the Principles of DNA Computation, Chinese J. Computers 24(9), 972-978 (2001)
[12] Lipton, R.J.: DNA solution of hard computational problems, Science 268(28), 542-545 (1995)
[13] Motwani Rajeev, Raghavan Prabhakar: Randomized Algorithms (Cambridge University Press, USA). (1995)
[14] Rohani Binti Abu Bakar, Junzo Watada, Witold Pedrycz: DNA approach to solve clustering problem based on a mutual order, BioSystems 91, 1-12 (2008)
[15] Russell Deaton, Max Garzon, John Rose, et al.: DNA Computing: A Review, Fundamenta Informaticae 30, 23-41 (1997)
[16] Saeed Arida: Contextualizing generative design, Thesis of Damascus University, (2004)
[17] Jin Xu, She-Min Zhang, Yue-Ke Fan, et al.: DNA Computer Principle, Advances and Di_culties (III): The Structure and Character of ”Data” in DNA Computing, Chinese Journal of Computers 30(6), 869-880 (2007)