OPTIMIZATION OF DNA ENCODING BASED ON COMBINATORIAL CONSTRAINTS

JINSONG LI, QIANG ZHANG*, RUI LI AND SHIHUA ZHOU
Liaoning Key Lab of Intelligent Information Processing
Dalian University
Dalian 116622, P. R. China
*Corresponding author: zhangq30@yahoo.com
Received October 2007; accepted January 2008

Abstract. The design of DNA sequences plays an important role in DNA computing. In this paper, we primarily focus on the combinatorial constraints of DNA sequences in DNA encoding, and then analyze their relationships. We propose the evaluative formulas based on those constraints and transform DNA sequences design into a multi-objective optimize problem. The artificial immune algorithm (AIA)/simulated annealing algorithm (SAA) is presented to solve the problem, and better DNA sequences than the previously known constructions are obtained.

Keywords: DNA encoding, Combinatorial constraints, Multi-objective optimize, AIA/SA algorithm

1. Introduction. DNA computing started in 1994 when Adleman solved a hard (NP-complete) computational problem with an exceptional method [1]. Because of the huge numbers of DNA molecules in a typical test tube, any method of computation based on DNA would seem to have potential for a massive parallelism, capacity, and power. This potential, however, is limited by the constraints imposed by the chemical of DNA.

DNA of higher species consists of two complementary chains twisted around each other to form a double helix. Each chain is a linear sequence of nucleotides, or bases—two purines, adenine (A) and guanine (G), and two pyrimidines, thymine (T) and cytosine (C). The purine bases and pyrimidine bases are Watson-Crick (WC) complements of each other, in the sense that

\[ \overline{A} = T, \overline{T} = A, \overline{C} = G, \overline{G} = C \]  

Although there are only four kinds of nucleotides in DNA strand, and two kinds of complement ways, base pairs with a wide variety of sequencing constitute the diversity of DNA molecule, and then influences DNA computing.

In DNA computation, the core reaction is the specific hybridization between DNA sequences or Watson-Crick complement, which directly influences the reliability of DNA computation with its efficiency and accuracy. However, false hybridization can occur because of the chemical characteristics of DNA molecules [2], which could cause wrong results in DNA computing. For example, as shown in Figure 1, false hybridization happened in 2, 3, 4, and 5.

The DNA encoding problem means trying to encode every bit of DNA code words in order to make the DNA strands hybridize with its complement specifically in bio-chemical reaction. But, actually, DNA sequence encoding is a very hard optimization problem. In DNA computing, DNA molecules as "data" couldn’t be generated stochastically, because if the sequences are not carefully chosen, DNA sequence may fold back onto itself, forming a secondary structure which completely inhibits the sequence from participating in the computational process, and how to select the length of DNA sequence according to the scale of problems needs to be considered. Hence, in accordance with the specific problem
to be solved, what kind of encoding is to be chosen for optimizing the DNA sequence is a fundamental problem in DNA computing.

So far, several papers have proposed different algorithms to design DNA sequences [5-15,20,21], for example, Deaton and Garzon introduced a new measure of hybridization likelihood based on Hamming distance and proposed a theory of error-preventing codes for DNA computing [5]. Marathe et al. proposed dynamic programming algorithms based on Hamming distance and free energy [6]. Frotos et al. proposed template-map method [7]. Hartemink et al. developed the program “SCAN” [8], Arita et al. developed a sequence design system using GA and a random generate-and-test algorithm [9]. Tanaka et al. developed a support system for sequence design using SA algorithms, and listed up some fitness criteria [10]. Deaton et al. proposed an evolution search method [11]. Soo-Yong Shin et al. developed an evolutionary sequence generation system to minimize the potential errors for DNA computing [12]. In [13,14], the authors used stochastic search algorithms to design codewords that are suitable for DNA computing. Rykov et al. proposed to use cyclic codes over to construct DNA codes [15]. In [20,21], Zhou et al. used the Particle Swarm Optimization to get the DNA sequence that for DNA computing.

In a sense, DNA encoding is essentially a multi-objective optimization problem. The modern intelligent optimization methods (Genetic Algorithm, Artificial Immune Algorithm, Particle Swarm Optimization, Ant Colony Optimization Algorithm et al) present effective ways to solve complex optimization problems. Various optimization algorithms in the DNA encoding are proved to be very good application. In Soo-Yong Shin’s paper, better sequences were generated by using GA algorithm to solve the multi-objective optimization problem [12]. In [13] Wang used GA/SA to get DNA sequences. However, in the traditional genetic algorithm, probability of crossover and mutation are constant, the individual with better fitness value and the population diversity may be damaged. So in this paper, we use Artificial Immune Algorithm (AIA) [17-18], in combination with simulated annealing algorithm to solve the multi-objective optimization problem, and we get a better result of the DNA sequence than the previously best known results [12,13], the DNA sequence with good structures can limit non-specific hybridzation, and have similar thermodynamic stability.

2. DNA Encoding and Its Constraints.

2.1. DNA encoding [5]. The DNA encoding problem can be described as follows: the encoding alphabet of DNA sequence is the set of nucleic acid bases $\Sigma = \{A, T, C, G\}$, and in the encoding set $Z$ of DNA strands whose length is $L$, $Z = \Sigma^L = \{< b_1, b_2, \cdots, b_n > : |b_i \in \Sigma, i = 1, 2, \cdots, L \} \text{ and } |Z| = 4^L$, search the subset $W$ of $Z$ which satisfies for $\forall x_i, x_j \in W, \tau(x_i, x_j) > k$, where $k$ is a positive integer, and $\tau$ is the expected criterion of evaluating the encoding, namely the encodings should satisfy constraint conditions.
2.2. DNA encoding constraints. Theoretically, the DNA encoding should satisfy two kinds of constraints, one is combinatorial constraints; the other is thermodynamic constraints.

2.2.1. Combinatorial constraints. Anyone of computing paradigm can be attributed to the information transmission and processing. Error-correcting codes in information theory effectively solved the problem of electronic computer encoding. Its mathematical basis is to measure distance between the two vertexes in the binary hypercube space by hamming distance. Due to the special nature of DNA Computing, we give their definition [6]:

In the following discussion, \( x_i (1 \leq i < m), x_j (1 \leq j < m) \) are supposed as the DNA sequence whose length is \( n \), and \( m \) denotes the count of DNA sequence in alphabet \( \Delta = \{A, C, G, T\} \). For convenience, DNA strand is oriented from the 5' to 3' end. \( x' \) whose orientation is the 3' to 5' end, is supposed as Watson-Crick complement of a strand \( x \). \( x' \) denotes the reverse of strand \( x \).

1) Hamming Distance: the number of position \( i \) at which the \( i \)th letter in \( x_i \) differs from the \( i \)th letter in \( x_j \).

2) Reverse Hamming Distance: the Hamming Distance between the sequence \( x_i \) and \( x'_j \). If \( x_j = q_1 q_2 \cdots q_n \), then \( x'_j = q_n q_{n-1} \cdots q_1 \).

3) Complement Hamming Distance: the Hamming Distance between the sequence \( x_i \) and \( x_c \). For binary sequences, their complement sequence is to change all "0" to "1" and "1" to "0", but in the DNA computing, the nucleotides in DNA strand must be changed into the nucleotides of their Watson-Crick Complement and meanwhile, the direction of DNA strand change from 5' → 3' to 3' → 5'.

4) Reverse Complement Hamming Distance Constraint: the Hamming Distance between \( x_c \) and \( x'_j \).

5) Continuity [12,22]: If one letter repeats continuously in DNA sequence, and then the DNA strand structure will become unstable, we should try to avoid in DNA encoding.

6) Hairpin structure [12]: the hairpin structure is caused by DNA sequence folding back onto itself due to self-hybridization. The hairpin structure is not expect in most ways of DNA computing, in our paper, one of goal of DNA encoding is try to reduce the number of its occurrences.

2.2.2. Thermodynamic constraints. 1) Free energy: Free energy is released energy in the transformation that the single stranded DNA molecules spontaneously transform from high-energy state to double stranded molecules of lower-energy state. Free energy change \( \Delta G \) is an important parameter evaluating the DNA molecular thermodynamic stability. \( \Delta G \) is usually calculated approximately by Nearest Neighbor Model parameters that defined in [19]:

\[
\Delta G = \theta + \sum_{i=1}^{n-1} w(b_i, b_{i+1})
\]

where \( \theta \) is modifiatory value, \( \omega \) is the specifically minus value of the sequence \( b_i b_{i+1} \).

2) Melting Temperature: Melting temperature is also important for efficiency and reliability of the DNA reaction, there are many factors that influence it., such as the component of DNA, the concentration of DNA, the pH of liquor and so on, the follows (3) is to calculated melting temperature in the nearest neighbor model with SantaLucia’s unified NN parameters [19]

\[
T_m = \Delta H^o / (\Delta S^o + R \ln C_t)
\]

3. Algorithm.
3.1. **Artificial immune algorithm (AIA).** The immune system is the basic and remarkable defence system against bacteria, viruses and other disease-causing organisms. It can produce millions of antibodies from hundreds of antibody genes and can protect animals which are infected by foreign molecules [17-18]. The Artificial Immune Algorithm (AIA) was inspired by the immune system. In AIA, the objective function and constraints operate as the antigens, and the solutions to the objective function operate as the antibodies. Similar to Genetic Algorithm (GA), AIA starts by creating antibodies randomly in a feasible space, and finally reaches the optimum via natural selection, crossover and mutation. Compared with GA, AIA has an affinity calculation function, which can describe the relationship not only between the antigen and the antibody but also between antibodies, which gives AIA the unique characteristic of guaranteeing the survival of variant offspring that can match the antigen better. The higher the affinity, the stronger the binding, and thus the better the immune recognition and response.

3.2. **Design of antigen.** For comparing with the previous works [12,13], we propose the evaluation function according to Section 2.

1) Hamming Distance Constraint (HDC): A large Hamming distance should be held between any two sequences.

\[ f_{\text{Hamming}}(i) = \min_{1 \leq j < m, j \neq i} \{ H(x_i, x_j) \} \]  

where \( f_{\text{Hamming}}(i) \) indicates the Hamming evaluation function of the \( i \)th individual.

2) Reverse Hamming Distance Constraint (RHDC):

\[ f_{\text{Reverse}}(i) = \min_{1 \leq j < m} \{ H(x_i, x_{\beta j}) \} \]

3) Reverse Complement Constraint (RCC):

\[ f_{\text{Reverse Complement}}(i) = \min_{1 \leq j < m} \{ H(x_i', x_{\beta j}') \} \]

4) Continuity Constraint (CC):

\[ f_{\text{Con}}(i) = -\sum_{j=1}^{n} (j - 1)N_j^{(i)} \]

where \( N_j^{(i)} \) denotes the number of times to which the same base appears \( j \)-times continuously in sequence.

5) Hairpin Constraint (HC):

\[ f_{\text{Hairpin}}(i) = \sum_{r=5}^{n-2-\text{pinlen}} \sum_{c=\text{pinlen}+[r/2]}^{n-\text{pinlen}+[r/2]} \text{Hairpin}(x_i, c) \]

where \( r \) is the minimum length to form hairpin ring, \( \text{pinlen} \) denotes the minimum sequence length of hybridization to form hairpin, \( \text{Hairpin}(x_i, c) \) is 1, when the reverse-complement distance of two sequence which sequence \( x_i \) is folded around the \( c \)th base is more than \( \text{pinlen}/2 \), otherwise is 0.

6) \( GC \) Content Constraint:

\[ f_{\text{GC}}(i) = -\lambda |GC(i) - GC(i)_{\text{defined}}| \]

where \( GC(i)_{\text{defined}} \) is the target value of \( GC \) content of DNA sequence \( x_i \), and \( GC(i) \) is the real \( GC \) content, \( \lambda \) is the parameter that is used to adjust the weight of the constraint and other constraints. In this paper, we use evaluation function of \( GC \) content to evaluate melting temperature, due to the (10). Where \( \text{Length} \) is the of length DNA sequence.

\[ T_m = 81.5 + 41 \times \text{RatioGC} - 500/\text{length} \]
7) Free energy Constraint: Besides the concentration of reactant, free energy is influenced by the component of DNA. Under the similar conditions, the more degree of hybridization, the more change of free energy, and hamming distance etc could reflect the degree of hybridization in some extent, so in our paper, the free energy is not considered.

We formulate the evaluation function as a maximum problem, and use the weighted sum to deal with the every evaluation function of constraints selected.

\[ f_j \in \{ f_{\text{Hamming}}(i), f_{\text{Reverse}}(i), f_{\text{Reverse Complement}}(i), f_{\text{Con}}(i), f_{\text{Hairpin}}(i), f_{\text{GC}}(i) \} \]  

\[ \text{fitness}(i) = \sum_{i=1}^{6} W_i f_i \]

3.3. Steps of algorithm. AIA introduces a new operator -immune operator based on the framework of the original genetic algorithm, and including new steps, one is inoculating bacterin which improves the fitness of the population and the other is immune choice which prevents the degenerateness of the population. But those steps can’t overcome the disadvantage of genetic algorithm which is difficult to break away from the local optimal value. The simulated algorithm (SA) can improve the premature convergence default in AIA algorithm. So in this paper, we use simulated algorithm when the antibodies update, and SA operator is designed as follows:

\[ P(\text{new} \rightarrow \text{old}) = \begin{cases} 1 & \text{if } \text{fitness(new)} \geq \text{fitness(old)} \\ \exp\{\lambda(\text{fitness(new)} - \text{fitness(old)})/T\} & \text{if } \text{fitness(new)} < \text{fitness(old)} \end{cases} \]

Steps of Algorithm:

Step1: Recognize the invasion of antigens, which corresponds to input data of the scheduling problem. Here multi-objective optimize problem is considered as the antigens, and IGA parameters are inputted.

Step2: Generate the antibodies through searching the memory antibodies of the problem, or generating randomly. In our experiment, the population size is 20.

Step3: Evaluate the fitness and calculates the affinities between antigen and antibody, antibody and antibody.

Step4: Chooses the antigen with larger affinity to input the memory cells and take place of the old antigen.

Step5: Generate antibody’s expectation, the antigen with lower expectation will be restrained and the antigen with higher expectation will be prompted.

Step6: The antibodies that will perform the next generation to optimization are proliferated by selection, crossover and mutation with certain probabilities. Then some more antibodies are generated from memory cells.

Step7: If the termination condition is not true, adjust the parameter in SA operator, go to Step 3, otherwise the system is over.

4. Result and Analysis.

4.1. Simulation results. AIA/SA mentioned above is implemented with Matlab 7.1. The AIA/SA parameters used in our example are: the population size is 20, the generation number is 300, DNA sequence length is 20, probability of crossover is 0.6, the mutation rate is 0.01, initial temperature and cooling schedule about SA are set to be 180 and 0.95, respectively, and the required individual count is 7. Figure 2 illustrates the results of simulation. From Figure 2, we can see that the algorithm has good convergence, and it could get optimal solutions with less generation times comparing with the algorithm in [13]
4.2. Comparisons. Table 1 shows the fitness value of DNA sequence based on combinatorial constraints. In our example, better fitness value is obtained by comparing with sequences in the previous works [12,13].

Table 1. The comparison of sequences based on Combinatorial Constraints

<table>
<thead>
<tr>
<th>DNA Sequence (5’-3’)</th>
<th>HDC</th>
<th>RHDC</th>
<th>RCC</th>
<th>CC</th>
<th>Hairpin</th>
<th>GC(%)</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DNA sequences in our system</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGCTGAGACGCTGTGTTACGA</td>
<td>13</td>
<td>15</td>
<td>15</td>
<td>-1</td>
<td>0</td>
<td>50</td>
<td>42</td>
</tr>
<tr>
<td>TATGACTCATGACGCTGACGC</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>41</td>
</tr>
<tr>
<td>CAGCTCTCGTACATCGACAAC</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>-1</td>
<td>0</td>
<td>50</td>
<td>41</td>
</tr>
<tr>
<td>CAGCGTGTCACTACGCGG</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>-1</td>
<td>-1</td>
<td>50</td>
<td>41</td>
</tr>
<tr>
<td>GATGCAAGACGACGTAATCA</td>
<td>13</td>
<td>14</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>GCTATATGCGACGACAGTAC</td>
<td>14</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>ATAGAGATCTCTACGCGCTG</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>-2</td>
<td>0</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td><strong>Sequences in Wang’s paper [13]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTATAGACGACGATGCACGCA</td>
<td>13</td>
<td>12</td>
<td>15</td>
<td>-1</td>
<td>0</td>
<td>50</td>
<td>39</td>
</tr>
<tr>
<td>ACAGACAGTGCTACAGGCACG</td>
<td>14</td>
<td>14</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>38</td>
</tr>
<tr>
<td>TGGTCCTGCACTACGTCTAGT</td>
<td>12</td>
<td>15</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>45</td>
<td>38</td>
</tr>
<tr>
<td>TACCGCCACACATGAAAGT</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>-2</td>
<td>0</td>
<td>50</td>
<td>37</td>
</tr>
<tr>
<td>ACAGACTGACGGACCCATG</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>-1</td>
<td>-2</td>
<td>50</td>
<td>37</td>
</tr>
<tr>
<td>ATAGCTGTACGCTGCTGTCAG</td>
<td>12</td>
<td>13</td>
<td>12</td>
<td>0</td>
<td>-1</td>
<td>50</td>
<td>36</td>
</tr>
<tr>
<td>TCTTCGCTATCTCTGCTGAG</td>
<td>12</td>
<td>13</td>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>50</td>
<td>36</td>
</tr>
<tr>
<td><strong>Sequences in Soo-Yong Shin’s paper [12]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAGTTAGATGTCACGTACAG</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>-1</td>
<td>-4</td>
<td>50</td>
<td>37</td>
</tr>
<tr>
<td>AGCGAGTAGGGGATATATC</td>
<td>14</td>
<td>12</td>
<td>13</td>
<td>-4</td>
<td>-1</td>
<td>50</td>
<td>34</td>
</tr>
<tr>
<td>TTAGGATTCCATGGCGCTC</td>
<td>13</td>
<td>13</td>
<td>11</td>
<td>-4</td>
<td>0</td>
<td>50</td>
<td>33</td>
</tr>
<tr>
<td>CCTGTCAACATTGACGCTCA</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>-3</td>
<td>-2</td>
<td>50</td>
<td>31</td>
</tr>
<tr>
<td>CCGCTCCATCTGATCGTTT</td>
<td>11</td>
<td>13</td>
<td>13</td>
<td>-5</td>
<td>-2</td>
<td>50</td>
<td>31</td>
</tr>
<tr>
<td>ATCGTACTCATGGCTCCCTAC</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>-3</td>
<td>-2</td>
<td>50</td>
<td>28</td>
</tr>
<tr>
<td>CTTCGCTGCTGATAACCTCA</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>-3</td>
<td>-1</td>
<td>50</td>
<td>28</td>
</tr>
</tbody>
</table>
5. **Conclusions.** In this paper, we have studied the combinatorial constraints of DNA encoding, and transformed the DNA encoding to multi-objective optimization problem. AIA/SA algorithms are presented to solve the optimization problem, and good sequences are obtained to improve the reliability of DNA computation. Finally, we have proved the efficiency of our algorithms by comparing with the previous works. In future work, we will examine ways to further improve our algorithm, and introduce the thermodynamic constraints in our algorithm.

**Acknowledgment.** This work was supported by National Natural Science Foundation of China (Grant Nos. 60403001, 60533010, 30740036) and Program from New Century Excellent Talents in University of China (NCET-06-0298)

**REFERENCES**


