The optimization of DNA encodings based on GA/SA algorithms*

Wang Wei[1], Zheng Xuedong[2], Zhang Qiang[1]** and Xu Jin[2]

(1. Liaoning Key Laboratory of Intelligent Information Processing, Dalian University, Dalian 116622, China; 2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Accepted on November 30, 2006

Abstract The design of DNA sequence plays an important role in improving the reliability of DNA computation. Proper constrained terms that DNA sequence should satisfy are selected, and then the evaluation formulas of each DNA individual corresponding to the selected constrained terms are proposed. The heuristic improved genetic algorithm (GA)/simulated annealing (SA) algorithm is presented to solve the multi-objective optimize problem, and the DNA sequence design system is developed. Furthermore, an example is illustrated to show the efficiency of our method given here.

Keywords: DNA encoding, multi-objective optimize, GA/SA algorithms.

In recent ten years, DNA computation[1] has been extensively researched as a new computation paradigm, because of its massive parallelism and huge storage which make it have the capability of solving the NP problems. In DNA computation, the core reaction is the specific hybridization between DNA sequences or the 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]. False hybridization in DNA computation process can be divided into two categories[3, 4]. One is false positive, i.e. non-specific hybridization occurs between a DNA strand and its Watson-Crick complement of a distinct DNA strand, in which case there are mismatches; the other is false negative, i.e. hybridization between a DNA strand and its complement does not take place as intended. The sufficient similarity between DNA molecules induces false positive results, while false negative results are caused for the reaction conditions and biochemical operations. Therefore, reasonable encoding design in advance is significant to avoid false hybridization and is helpful to improve the reliability of DNA computation.

The 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. Hence, the goal of encoding is mainly to minimize the similarity distance of the various DNA sequences by using some convenient similarity measure. So as a necessary criterion for reliable DNA computation, the minimum Hamming distance[2] and H-measure based on Hamming distance[5] between DNA code words were proposed to define the distance. And various algorithms and methods for the reliable DNA sequence design were presented based on the two distance measures. For example, Marathe et al. proposed dynamic programming algorithms based on Hamming distance[6]. Frotos et al. proposed template-map method[7]. Hartemink et al. developed the program "SCAN", and presented method of the encodings design based on some constraints[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 the previous work, DNA sequence design can be considered as a numerical optimization problem that satisfying constraints based on knowledge of sequence design. Among the optimization methods, the

* Supported by National Natural Science Foundation of China (Grant Nos. 60403001 and 60533010) and Program for New Century Excellent Talents in University

** To whom correspondence should be addressed. E-mail: zhangq26@126.com

GA algorithm is probably the most popular method of parameter optimization for a problem which is difficult to mathematically formalize. In Soo-Yong Shin's paper, better sequences were generated by using GA algorithm to solve the multi-objective optimization problem[12]. However, in the previous work, a holistic disposal was adopted to deal with every constraint, and in every generation the performance of each individual was evaluated by the fitness function of whole individuals in population. At the same time the inherent default, convergent in local point in GA algorithm, was not considered. Corresponding to it, in this paper, the evaluation formulas of every DNA individual corresponding to the selected constraint conditions are proposed, and the heuristic improved GA/SA algorithms are presented.

1 Design criterions

The encoding problem can be described as follows[13]: the encoding alphabet of DNA sequence is the set of nucleic acid bases A, T, G, C, and in the encoding set S of DNA strands whose length is N, search the subset C of S which satisfies for $\forall x_i, x_j \in C$, $\tau(x_i, x_j) \geqslant k$, where k is a positive integer, and τ is the expected criterion of evaluating the encoding, namely the encodings should satisfy constraint conditions. Theoretically, the encoding should satisfy two kinds of constraints: combinatorial constraints and thermodynamic constraints.

On the first kind of constraints in the DNA encoding problem, there are some constraint terms based on some distance measure as follows at present: Hamming constraint, reverse constraint, reverse-complement constraint, H-measure constraint, similarity constraint, continuity constraint, 3′-end H-measure constraint, and hairpin constraint. Obviously, regarding the first kind of constraints the DNA encoding problem can be translated to the multi-objective optimization problem, which means that the encodings should satisfy many constraints at the same time. However, in the various constraints mentioned above, some of them may overlap each other, so reasonable selection of every constraint term is very important.

As for the second kind of constraints, in the case that some conditions, such as temperature, pressure and so on, have been given, the free energy (ΔG) of the DNA double helix structure is a reliable way to measure the relative stability of DNA molecule. While the measures closely relating with free energy are melting temperature, and the proportion of character G and C in DNA sequence. Therefore, it is reasonable to abstract the second kind of constraints as the optimization problem of free energy and GC content. At present, the mostly used methods to calculate melting temperature are Wallace 2—4 rule, the GC% method and the nearest neighbor model[12]. In practice, the GC content is supposed to be 50% generally.

2 Design strategies

In this paper, six criterions based on Hamming distance measure are selected from various constraint conditions mentioned above to optimize DNA sequence. Since it is reported to be effective in laboratory experiments, the Hamming distance measure but not H-measure is used here[14]. Based on the evaluation terms listed up by Tanaka et al.[10] and Shin et al.[12], the fitness evaluation function corresponding to every DNA individual evaluation criterion was proposed, furthermore, the optimization problem was solved with SA/GA algorithms. In the following discussion, $x_i (1 \leqslant i \leqslant m)$, $x_j (1 \leqslant j \leqslant m)$ are supposed as the DNA sequence whose length is n, and m denotes the count of DNA sequence in population. For convenience, DNA strand x 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^R denotes the reverse of strand x.

2.1 Evaluation functions

Hamming constraint[14]: A large Hamming distance should be held between any two sequences. The evaluation function of the constraint is described as Eq.(1), where $f_{\text{Hamming}}(i)$ indicates the Hamming evaluation function of the ith individual in evolutionary population,

$$f_{\text{Hamming}}(i) = \min_{1 \leqslant j \leqslant m,\, j \neq i} \{H(x_i, x_j)\} \tag{1}$$

Reverse constraint[6]: The Hamming distance between x_i and x_j^R should not be less than a certain parameter d, i.e. $H(x_i, x_j^R) \geqslant d$.

$$f_{\text{Inverse}}(i) = \min_{1 \leqslant j \leqslant m} \{H(x_i, x_j^R)\} \tag{2}$$

Reverse-complement constraint[6]: A sequence should not hybridize with the reverse-complement of other sequences.

$$f_{\text{Inverse-Comple}}(i) = \min_{1 \leqslant j \leqslant m} \{ H(x_i', x_j^R) \} \quad (3)$$

Continuity constraint[10]: If the same base appears continuously, the structure of DNA will become unstable. The evaluation function is described as follows:

$$f_{\text{Con}}(i) = -\sum_{j=1}^{n} (j-1) N_j^{(i)} \quad (4)$$

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

GC Content constraint[9]: GC content affects the chemical properties of DNA sequence.

$$f_{\text{GC}}(i) = -\lambda \mid GC^{(i)} - GC_{\text{defined}}^{(i)} \mid \quad (5)$$

where $GC_{\text{defined}}^{(i)}$ is the target value of GC content of DNA sequence x_i, and $GC^{(i)}$ is the real GC content, λ is the parameter that is used to adjust the weight of the constraint and other constraints.

Hairpin constraint: In general, the hairpin structure formation is not desirable, because it can hybridize itself, and which may cause a secondary structure. Thus the following evaluation term was proposed by Shin et al.[12]

$$f_{\text{Hairpin}}(i) = \sum_{r=5}^{(n-2*pinlen)} \sum_{c=pinlen+\lceil r/2 \rceil}^{n-pinlen-\lfloor r/2 \rfloor} Hairpin(x_i, c) \quad (6)$$

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

T_m **constraint**: Melting temperature is also an important factor for the efficiency of the DNA reaction. The GC% method was adopted in the paper, and the equation was described as follows, where the Length is the length of DNA sequence.

$$T_m = 81.5 + 41 * RatioGC - 500/Length$$

Obviously, the GC content is the only argument of the melting temperature, hence the T_m constraint will not be considered as the optimization object in this paper.

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

$$f_j \in \{ f_{\text{Hamming}}(i), f_{\text{Inverse}}(i), f_{\text{Inverse-Comple}}(i) \\ f_{\text{Hairpin}}(i), f_{GC}(i), f_{\text{Con}}(i) \}$$

$$Fitness(i) = \sum_{j=1}^{6} w_j f_j \quad (7)$$

where w_j is the weight of the jth constraint, here, we set it to be 1.

2.2 Design of algorithms

SA/GA algorithms are the algorithms that SA algorithm is combined with GA algorithm. The SA algorithm can improve the premature convergence default in GA algorithm. In this paper, fitness function is proposed corresponding to each individual in population. So, in essence, the optimization problem means that the required individuals in population which have a high fitness value are selected from the every generation colony, when evaluation does not reach convergence or designated generation. In our algorithm, selection operator, crossover operator, mutation operator, inverse operator, and SA operator are carefully designed. In the selector operator, $(\mu+\lambda)$ strategy is adopted, and SA operator is designed as follows:

$$P(\text{new} \rightarrow \text{old}) = \begin{cases} 1 & fitness(\text{new}) \geqslant fitness(\text{old}) \\ \exp(\lambda(fitness(\text{new}) - fitness(\text{old}))/T) & fitness(\text{new}) < fitness(\text{old}) \end{cases} \quad (8)$$

where $fitness$(new) denotes the fitness value of a new individual after selection, crossover, mutation, and inverse operator. $fitness$(old) denotes the fitness value of an old individual before selection, crossover, mutation, and inverse operator. P(new → old) denotes the received probability of a new individual, λ is the parameter in SA operator, in simulation, and λ is set to be 20.

Steps of GA/SA algorithms solving the sequence design are as follows:

Step 1: Set parameters, and initialize population randomly.

Step 2: Calculate the fitness value of every individual in population by descending sort.

Step 3: Select required individuals in population to enter the next generation directly by adopting the elite strategy. The set of individuals is called set C, and the rest are saved to an array called old array. The individuals in an old array after multiple point

crossover, single point mutation, two point inverse are saved as an array denoted as a new array.

Step 4: Adopt the $(\mu+\lambda)$ strategy to select the individual in the old array and new array, where every individual's fitness value is calculated between the individuals in set C and itself that has not been signed. The highest fitness value individuals are selected separately in the new array and old array, then, the individual is selected from the two individuals in terms of Meteopolis rule, and it is signed and added with the set C. Step 4 is repeated till the count of the members in set C satisfies the size of population.

Step 5: If the termination condition is not true, adjust the parameter in SA operator, go to step 2, otherwise go to step 6.

Step 6: End.

3 Simulation results

GA/SA algorithms mentioned above are implemented in VISUAL BASIC language. The GA/SA algorithms 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 0.05, probability of inversion is 0.4, initial temperature and cooling schedule about SA are set to be 100 and 0.95, respectively, and the required individual count is 7. Fig. 1 illustrates the results of simulation, where the fitness value of the 7th individual which has the lowest fitness value among the required seven individuals is the value calculated in the whole population. Better convergence performance after 225 generation is shown in Fig. 1.

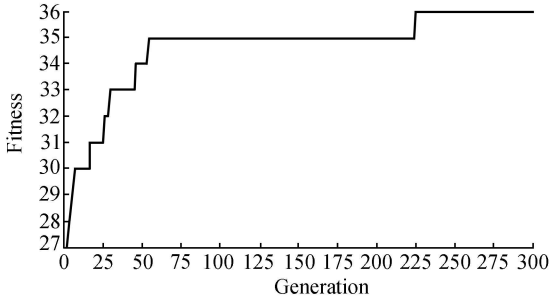

Fig. 1. Evolution graph of SA/GA algorithms.

3.1 Comparison in DNA sequences

To evaluate the performance of algorithms, seven good encodings are generated, and are compared with the encodings in previous work in various rules (i.e. various constraint terms are adopted).

3.1.1 DNA sequence comparison based on Hamming distance

Table 1 shows the fitness value of DNA sequences based on Hamming distance. In our example, better fitness value is obtained by comparing with sequences given in Shin's paper[12]. Here, we calculate every object function based on Hamming distance of previous sequences.

Table 1. The comparison of sequences based on Hamming rule

DNA Sequence (5′—3′)	Distance[a)]	Distance[b)]	Distance[c)]	GC(%)	Continuity	Hairpin	Fitness
DNA sequences in our system							
ACTATAGACAGCATGCCGCA	13	12	15	50	−1	0	39
ACAGACAGTGCTACAGCACG	14	14	12	55	0	0	38
TGATCTGCACATGTCTAGTG	12	15	13	45	0	0	38
TACGCGCACACATGAAAGTG	14	13	12	50	−2	0	37
ACGACTGACGTAGCCATGAT	13	13	14	50	−1	−2	37
ATAGCTGTACGTCGTGTCAG	12	13	12	50	0	−1	36
TCTTGCGTATCTCTGCTGAG	12	13	12	50	−1	0	36

To be continued

Continued

DNA Sequence (5′—3′)	Distance[a)]	Distance[b)]	Distance[c)]	GC(%)	Continuity	Hairpin	Fitness
Sequences in Soo-Yong Shin's paper[12]							
GAGTTAGATGTCACGTCACG	15	14	13	50	−1	−4	37
AGGCGAGTATGGGGTATATC	14	12	13	50	−4	−1	34
TTATGATTCCACTGGCGCTC	13	13	11	50	−4	0	33
CCTGTCAACATTGACGCTCA	11	11	14	50	−3	−2	31
CGCTCCATCCTTGATCGTTT	11	13	13	50	−5	−2	31
ATCGTACTCATGGTCCCTAC	11	10	12	50	−3	−2	28
CTTCGCTGCTGATAACCTCA	11	10	11	50	−3	−1	28

a) Hamming distance; b) reverse distance; c) reverse-complement distance.

3.1.2 DNA sequence comparison based on H-measure distance

The difference of constraint terms selected mainly manifests in the Hamming distance or in the H-measure used. In this paper, the evaluation functions of H-measure constraint and similarity constraint based on H-measure are defined as follows:

$$f_{H\text{-measure}}(i) = n - \max_{1\leqslant j\leqslant m}\ \max_{-n<k<n}\{n - H(x_i, \sigma^k(x_j'))\} \quad (9)$$

$$f_{\text{Similarity}}(i) = n - \max_{1\leqslant j\leqslant m,\ i\neq j}\ \max_{-n<k<n}\{n - H(x_i, \sigma^k(x_j))\} \quad (10)$$

where $\sigma^k(x_j)$ indicates the right (left) shift in case of $k>0$ ($k<0$), k denotes the number of the shift.

Table 2 shows that our sequences have good performance in H-measure and similarity object function, and our sequences also have the good fitness value.

Table 2. The comparison of sequences based on H-measure rule

DNA Sequence (5′-3′)	Distance[a)]	Similarity	GC(%)	Continuity	Hairpin	Fitness
DNA sequences in our system						
ACTATAGACAGCATGCCGCA	12	12	50	−1	0	23
ACAGACAGTGCTACAGCACG	12	12	50	0	−1	23
TGATCTGCACATGTCTAGTG	10	13	50	−1	−2	20
TACGCGCACACATGAAAGTG	10	12	45	0	0	20
ACGACTGACGTAGCCATGAT	10	12	50	−2	0	20
ATAGCTGTACGTCGTGTCAG	10	12	55	0	0	20
TCTTGCGTATCTCTGCTGAG	10	10	50	−1	0	19
Sequences in Shin's paper[12]						
GAGTTAGATGTCACGTCACG	11	13	50	−4	−1	19
AGGCGAGTATGGGGTATATC	11	12	50	−4	0	19
TTATGATTCCACTGGCGCTC	11	11	50	−3	−1	18
CCTGTCAACATTGACGCTCA	10	12	50	−1	−4	17
CGCTCCATCCTTGATCGTTT	11	11	50	−3	−2	17
ATCGTACTCATGGTCCCTAC	10	9	50	−3	−2	14
CTTCGCTGCTGATAACCTCA	10	11	50	−5	−2	14

a) Hamming distance

4 Conclusion

In this paper, some constraint terms based on Hamming distance are selected from various constraint terms, and then the selected terms are transformed to multi-objective optimization problem. GA/SA algorithms are proposed to solve the optimization problem, and good sequences are obtained to improve the reliability of DNA computation. Finally, the feasibility and efficiency of our method are proved by comparing our sequences with other sequences.

References

1 Gao L, Xu J and Zhang JY. A survey of DNA computing. Acta Electronnica Sinica, 2001, 7: 973—975

2 Deaton R, Murphy RC, Garzon M, et al. Good encodings for DNA-based solutions to combinatorial problems. In: Proceedings of 2nd DIMACS Workshop on DNA Based Computers, 1996, 159—171
3 Deaton R and Garzon M. Thermodynamic constraints on DNA-based computing. In: Computing with Bio-Molecus: Theory and Experiments, Singapore: Spinger-Verlag, 1998, 138—152
4 Deaton R, Franceschetti DR, Garzon M, et al. Information transfer through hybridization reaction in DNA based computing. In: Proceedings of the Second Annual Conference, 1997, 463—471
5 Garzon M, Neathery P, Deaton R, et al. A new metric for DNA computing. In: Proceedings of the 2nd Annual Gentic Programming Conference, 1997, 472—487
6 Marathe A, Condon AE and Com RM. On combinatorial DNA word design. In: Proceedings of 5th DIMACS Workshop on DNA Based Computers, 1999, 75—89
7 Frutos AG, Liu QH, Thiel AJ, et al. Demonstration of a word design strategy for DNA computing on surface. Nucleic Acids Res, 1997, 25: 4748—4757
8 Hartemink AJ, Gifford DK and Khodor J. Automated constraint-based nucleotide sequence selection for DNA computation. In: Proceedings of 4th DIMACS Workshop on DNA Based Computers, 1998, 227—235
9 Arita M, Nishikawa A, Hagiya M, et al. Improving sequence design for DNA computing. In: Proceedings of Genetic and Evolutionary Computation Conference, 2000, 875—882
10 Tanaka F, Nakatsugawa M, Yamamoto M, et al. Developing support system for sequence design in DNA computing. In: Preliminary Proceedings of 7th international Workshop on DNA-Based Computers, 2001, 340—349
11 Deaton R, Murphy RC, Rose JA, et al. A DNA based implementation of an evolutionary search for good encodings for DNA computation. In: Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, 1997, 267—272
12 Shin SY, Kim DM, Lee IH, et al. Evolutionary sequence generation for reliable DNA computing. In: Congress on Evolutionary Computation, 2002, 1: 79—84
13 Liu WB, Wang SD and Xu J. Research on the encoding method of DNA computing. Computer Engineering and Applications (in Chinese), 2003, 27: 118—121
14 Deaton R, Garzon M, Murphy RC, et al. Reliability and efficiency of a DNA-based computation. Physical Review Letters, 1998, 80: 417—420