Multi-objective carrier chaotic evolutionary algorithm for DNA sequences design*

Xiao Jianhua, Xu Jin, Geng Xiutang and Pan Linqiang**

(Key Laboratory of Image Processing and Intelligent Control, Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Accepted on June 14, 2007

Abstract DNA computing is a new vista of computation, which is of biochemical type. Since each piece of information is encoded in biological sequences, their design is crucial for successful DNA computation. DNA sequence design is involved with a number of design criteria, which is difficult to be solved by the traditional optimization methods. In this paper, the multi-objective carrier chaotic evolution algorithm (MCCEA) is introduced to solve the DNA sequence design problem. By merging the chaotic search base on power function carrier, a set of good DNA sequences are generated. Furthermore, the simulation results show the efficiency of our method.

Keywords: DNA computing, carrier chaotic search, multi-objective evolutionary algorithm, DNA sequences design.

DNA computing is a new computation paradigm, where the information is encoded by DNA sequences. In 1994, Adleman[1] firstly demonstrated the feasibility of solving NP-complete problems by DNA molecules. Because DNA computing has many good characteristics such as massive parallelism, exceptional energy efficiency, and huge storage density, DNA computing has been extensively investigated, for example, it has been used to solve NP-complete problems[2—4]. Since the information in DNA computing is encoded by DNA sequences, the design of DNA sequences is crucial for successful DNA computation. For a set of DNA sequences being effective in DNA computing, they must fulfill a number of combinatorial and thermodynamic constraints, which is difficult to be solved by the traditional optimization methods.

Many researchers have proposed various algorithms and methods for the reliable DNA sequence design. For example, Marathe et al.[5] proposed a dynamic programming approach; Frutos et al.[6] proposed a template strategy to select a huge number of dissimilar sequences; Aritha et al.[7] introduced genetic algorithm into DNA sequences design system and proposed a random generate-and-test algorithm; Tanaka et al.[8] applied simulated annealing to optimize the set of DNA sequences; Deaton et al.[9] proposed DNA sequences design algorithm based on evolutionary search method; Cui et al.[11] proposed DNA sequences design algorithm based on the PSO optimization; Wang et al.[15] developed GA/SA algorithm for DNA sequences design.

Chaos is a universal nonlinear phenomenon, and has the characteristics of the ergodicity, randomicity and regularity. In this paper, the carrier chaotic search was merged into multi-objective evolutionary algorithm, and a multi-objective carrier chaotic evolutionary algorithm (MCCEA) for designing DNA sequences was developed. In each generation of MCCEA algorithm, carrier chaotic search was performed on the copy of several individuals chosen randomly from the external archive to obtain new non-dominated solutions. More non-dominated solutions were produced by ergodic regularity of chaos. Compared with the traditional algorithm, such as PSO algorithm[11], GA/SA algorithms[15], and weight methods[12], our algorithm not only avoids the difficulty of selecting the proper weight values for each criterion and escaping from local optimal solution, but also produces a set of alternative solutions but a single optimal solution. The simulation results show that the comprehensive performance of multi-objective carrier chaotic evolutionary algorithm is improved by merging chaos

* Supported by National Natural Science Foundation of China (Grant Nos. 60373089, 60674106, 30570431, and 60533010), the Program for New Century Excellent Talents in University (Grant No. NCET-05-0612), the Ph. D. Programs Foundation of Ministry of Education of China (Grant No. 20060487014)), and the Chenguang Program of Wuhan (Grant No. 200750731262)

** To whom correspondence should be addressed. E-mail: lqpan@mail.hust.edu.cn

in MOEA algorithm.

1 Constraints formulation in DNA sequences design

In DNA computing, DNA sequences should not form any undesired secondary structures and must meet some physical, chemical and logical constraints in order to avoid mishybridization. Generally, the constraints such as H-measure, continuity, melting temperature, and GC content, need to be considered in the design of good DNA sequences. In this section, first, the various constraints for DNA sequences design will be described in detail. Then DNA sequences design problem will be formulated as a multi-objective optimization problem.

In the following context, $x_i (1 \leqslant i \leqslant m)$, $x_j (1 \leqslant j \leqslant m)$ are used to denote the DNA sequences with length n, and m is the cardinality of a set of DNA sequences. For convenience, DNA sequence x is oriented from $5'$ to $3'$ end, and the reverse orientation is the $3'$ to $5'$. Waston-Crick complement of a sequence x is denoted by $\bar{x}$, and x^R denotes the reverse sequence of sequence x.

1.1 Design constraints

1.1.1 Hamming distance constraint

Hamming distance $H(x_i, x_j)$ of two DNA sequences x_i and x_j is the number of corresponding places where two characters are different. In DNA sequences design, Hamming distance will ensure that each sequence is as nonsimilar as possible. The evaluation function $f_{\mathrm{Ham}}(i)$ of the Hamming distance constraint is defined as

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

1.1.2 GC content constraint

The GC content is the percentage of G and C in a DNA sequence. GC content affects the chemical properties of DNA and can reduce the probability of occurring non-specific hybridization effectively. The evaluation function $f_{\mathrm{GC}}(x_i)$ of the GC content constraint is described as follows:

$$f_{\mathrm{GC}}(x_i) = [\mathrm{GC}_{\mathrm{gen}}(x_i) - \mathrm{GC}_{\mathrm{tar}}(x_i)]^2 \tag{2}$$

where $\mathrm{GC}_{\mathrm{tar}}(x_i)$ is the target GC content of DNA sequence x_i, and $\mathrm{GC}_{\mathrm{gen}}(x_i)$ is the GC content of the generated sequence.

1.1.3 Tm constraint

Melting temperature is one of the most important factors for the DNA sequence design. There are many methods to calculate melting temperature such as the GC% method[13], and the nearest neighbor mode[14]. We use the GC% method to calculate melting temperature in this study. The evaluation function $f_{\mathrm{Tm}}(x_i)$ of melting temperature is defined as

$$f_{\mathrm{Tm}}(x_i) = [\mathrm{Tm}_{\mathrm{tar}}(x_i) - \mathrm{Tm}_{\mathrm{gen}}(x_i)]^2 \tag{3}$$

$$\mathrm{Tm}(x_i) = 81.5 + 16.6 \times \log_{10}\left(\frac{[\mathrm{salt}]}{1.0 + 0.7 \times [\mathrm{salt}]} \right) + 41 \times \mathrm{GC}\% - \frac{500}{|x_i|} \tag{4}$$

where $\mathrm{Tm}_{\mathrm{tar}}$ is the target melting temperature, and $\mathrm{Tm}_{\mathrm{gen}}$ is the target melting temperature of the generated sequence x_i, [salt] is salt concentration and $|x_i|$ is the length of DNA sequence x_i.

1.1.4 Reverse constraint

The Hamming distance $H(x_i, x_j^R)$ between x_i and x_j^R should not be lower than a given parameter. The formulation $f_{\mathrm{Rev}}(x_i)$ of the reverse constraint is defined as follows[15]:

$$f_{\mathrm{Rev}}(x_i) = \min_{1 \leqslant j \leqslant m} \{ H(x_i, x_j^R) \} \tag{5}$$

1.1.5 Continuity constraint

In a sequence, if the same bases occur continuously, it may cause unexpected biological structures. Continuity is often used to describe the degree of successive occurrence of the same base in a sequence. The formulation is defined as follows[16]:

$$F_{\mathrm{con}}(\Sigma) = \sum_{i=1}^{n} f_{\mathrm{con}}(x_i), f_{\mathrm{con}}(x) = \sum_{i=1}^{L-t+1} \sum_{\alpha \in \Lambda} T(C_\alpha(x, i), t)^2 \tag{6}$$

$$C_\alpha(x, i) = \begin{cases} c, & \text{if there is } c \text{ such that } x_i \neq \alpha, \\ & x_{i+j} = \alpha, \text{ for } 1 \leqslant j \leqslant c, \\ & x_{i+j+1} \neq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$T(i, j) = \begin{cases} i, & \text{if } i > j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $C_\alpha(x, i)$ is the number of the ith base α occurring continuously in DNA sequence x, and t is the target continuity.

1.1.6 Reverse complement constraint

Reverse complement constraint can reduce the probability of occurring non-specific hybridization with the reverse-complement of other sequences. The formulation $f_{RC}(x_i)$ of reverse complement constraint is defined as

$$f_{RC}(x_i) = \min_{1 \leqslant j \leqslant m} \{H(x_i, \bar{x}_j^R)\} \tag{9}$$

1.2 Multi-objective problem for DNA sequence design

DNA computing relies on biochemical reactions of DNA molecules and may result in incorrect or undesired secondary structures. Therefore, DNA sequences must meet some physical, chemical and logical constraints in order to avoid mishybridization. Generally, the constraints such as H-measure, continuity, melting temperature, and GC content, need to be considered and each constraint function needs to be optimized simultaneously. Obviously, the optimization problem of the constraint function can be formulated as multi-objective optimization problem. Formally, the DNA sequences design problem can be written as follows:

Optimize

$$\{f_{Ham}(x_i), f_{Rev}(x_i), f_{GC}(x_i), f_{Con}(x_i), f_{Tm}(x_i), f_{RC}(x_i)\} \tag{10}$$

The next section will describe in detail how to solve the multi-objective problem for DNA sequence design by multi-objective carrier chaotic evolutionary algorithm.

2 Multi-objective carrier chaotic evolutionary algorithms

2.1 Carrier chaotic optimization

Chaos is a kind of nonlinear phenomenon that widely occurs in nature. Because of the characteristics of chaotic motion such as the ergodicity, randomicity and regularity, the chaotic optimization can get rid of the local optimal solution. So the chaotic search has been introduced into the various optimization algorithms, such as chaotic neural network[17], chaotic simulated annealing algorithm[18], and mutative scale chaotic optimization algorithm[19]. In this paper, we will introduce chaotic search into the multi-objective evolutionary algorithm, which is used to solve DNA sequence design problem.

The mathematical expression of the logistical mapping is given as follows:

$$x_{n+1} = \mu \times x_n(1 - x_n), \quad \text{for } n = 0, 1, \cdots$$
$$\mu \in [0, 4], \quad x_n \in [0, 1] \tag{11}$$

where μ is the growth rate, and x_0 is the initial value. If $\mu = 4$, the system is of fully chaotic state.

The Logistic mapping has the advantage of ergodicity, but the probability density of logistic mapping orbit distributes ununiformity, and chaos searches mainly on the edge of the searching field. So power function carrier is proposed as follows:

$$z_n = \begin{cases} x_n^p, & \text{if } x_n \in [0, a] \\ x_n, & \text{if } x_n \in [a, b] \\ x_n^q, & \text{if } x_n \in [b, 1] \end{cases} \tag{12}$$

where $0 < a < b < 1$, $0 < p < 1$, $q > 1$, x_n is the chaotic variable produced by Eq. (11), and z_n is a new chaotic variable produced by power function. In this study, a is 0.35, b is 0.7, p is 0.6, and q is 3.

Obviously, z_n also has the characteristics of the ergodicity in [0, 1], and it is proved that the power function carrier can greatly enhance the global and local searching ability of the chaotic optimization[20].

2.2 Multi-objective evolutionary algorithm

Evolutionary algorithms have been proved to be well suited for optimization problems with multiple objective functions[21—23]. In the published literature on multi-objective evolutionary algorithms, the strength Pareto evolutionary algorithm (SPEA)[24] has showed good comprehensive performance. SPEA algorithm introduces elitism by explicitly maintaining an external population. At every generation, newly found nondominated solutions are compared with the existing external population and the resulting nondominated solutions are preserved.

The strength Pareto evolutionary algorithm is outlined in Fig. 1. Its basic procedure is as follows: (i) Creating new populations and marking non-dominated individuals, the external Pareto set is updated. (ii) If the number of externally stored Pareto solutions exceeds a given maximum, a reduced representation is computed by clustering. (iii) Select next generation individual from population and external archive by binary tournament strategy. (iv) Crossover and mutation are applied to the population as usual. For more notions on the strength Pareto

evolutionary algorithm, please refer to Ref. [24].

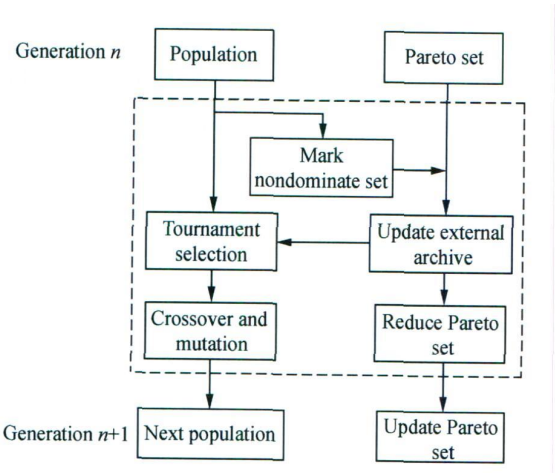

Fig. 1. Flowchart of the strength pareto evolutionary algorithm.

In the next section, carriers chaotic search is introduced into strength Pareto evolutionary algorithm to get more non-dominated solutions and improve the globally searching performance of algorithm. Details of the multi-objective carrier chaotic evolutionary algorithm will be described.

2.3 Multi-objective carrier chaotic evolutionary algorithm

In this study, chaotic search is merged into the strength Pareto evolutionary algorithm, and a multi-objective carrier chaotic evolutionary algorithm (MCCEA) for designing DNA sequences is developed. In each generation of MCCEA algorithm, carrier chaotic search is performed on the copy of several individuals chosen randomly from the external archive to obtain new non-dominated solutions. More non-dominated solutions are produced by ergodic regularity of chaos. Furthermore, due to the characteristics of the ergodicity, randomicity and regularity of chaos, the algorithm also escapes from the local optimal solution. The detailed procedure of the algorithm is given as follows:

Step 1. (Initialize population) Create new population.

Step 2. (Mark non-dominated individuals) Mark nondominated individuals. All non-dominated individuals in the population are copied to external archive.

Step 3. (Reduce Pareto set) If the number of externally archive Pareto solutions exceeds a given maximum, reduce the non-dominated set in external archive by clustering, otherwise go to step 4.

Step 4. (Binary tournament selection) Select next generation individual from population and external archive by binary tournament strategy.

Step 5. (Crossover and mutation) Execute crossover operator and mutation operator for population selected.

Step 6. (Update external archive) update external archive.

Step 7. (Carrier chaotic search) Select individuals randomly from external archive, then perform chaotic search on the copy of select individuals, obtain new nondominates solutions, and update external archive.

Step 8. (Termination) If the termination condition is not true, go to step 2, otherwise, go to step 9.

Step 9. End.

3 Simulation results

3.1 Algorithm parameters

In the above section, the multi-objective carrier chaotic evolution algorithm is proposed to select good DNA sequences. In the simulation, DNA sequences of length 20-mer are considered, and the bases "A, C, G, T" are mapped to 0, 1, 2, 3, respectively. In this way, a DNA sequence can be represented by a decimal number corresponding to this number. For example, a 20-ber DNA sequence CTAGCTAGAACGCGCTTCTT can be represented by the number sequence 13021302001212133133.

Multi-objective carrier chaotic evolutionary algorithm is implemented with MATLAB 7.0. The algorithms parameters used in our example are: the population size is 100, the maxgeneration number is 200, DNA sequence length is 20, probability of crossover and mutation rate is 0.7 and 0.03, respectively, salt concentration is 0.1 mol/L, and the max number of external non-dominated set is 50.

3.2 Results and analyses

In the simulation, the generated sequences and their objective values such as Hamming distance, Continuity, GC content, and so on, are listed in Table 1. The sequences and objective values in Table 2 are generated by Soo-Yong Shin's algorithm[16].

Table 1. DNA sequences and objective values in MCCEA algorithm

DNA sequences	Hamm distance	Reverse distance	Reverse complement distance	GC content	Continuity	Tm
TGAGCCGGAGTGTCAGGAAG	14	12	13	60	0	64.0122
CCGTCTGGACGTAGTAAGCT	13	12	13	55	0	61.9622
GGATGGAATGGAGAGCCGTA	12	13	12	55	0	61.9622
CTAGCTAGAACGCGCTTCTT	13	12	13	50	0	59.9122
AACCGCACAAGTCGCAATAT	13	13	12	45	0	57.8622
GTCTGACAGTACGAGACCGC	12	12	12	60	0	64.0122
GTTAGTTCCGAACCTCAGCG	14	12	12	55	0	61.9622

Table 2. DNA sequences and objective values in Soo-Yong Shin's algorithm

DNA sequences	Hamming distance	Reverse distance	Reverse Complement distance	GC content	Continuity	Tm
CTTCGCTGCTGATAACCTCA	11	10	11	50	0	59.9122
ATCGTACTCATGGTCCCTAC	11	10	12	50	9	59.9122
GAGTTAGATGTCACGTCACG	15	14	13	50	0	59.9122
AGGCGAGTATGGGGTATATC	14	12	13	50	16	59.9122
TTATGATTCCACTGGCGCTC	13	13	11	50	0	59.9122
CGCTCCATCCTTGATCGTTT	11	13	13	50	9	59.9122
CCTGTCAACATTGACGCTCA	11	11	14	50	0	59.9122

To evaluate the performance of algorithm, the averages of objective values from Tables 1 and 2 are calculated, which are shown in Fig. 2. Where the blue columns denote the averages calculated from Table 1, and the brown columns denote the averages calculated from Table 2. From Fig. 2, we can find that the DNA sequences generated by multi-objective carrier chaotic evolutionary algorithm have greater (reverse) Hamming values than the DNA sequences from Ref. [16]. It implies that DNA sequences generated by our algorithm are more nonsimilar, and can reduce the probability to hybridize with its noncomplementary sequences. Furthermore, the second structure of the DNA sequences generated by our algorithm is more restrained because of the low continuity. In general, the DNA sequences generated by our algorithm have better quality than that generated by Soo-Yong Shin's algorithm[16].

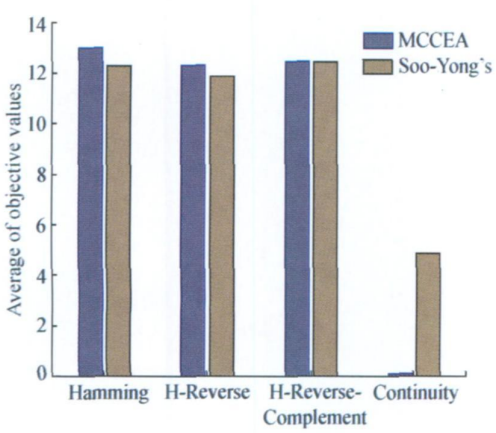

Fig. 2. Comparison of average objective values between Shin's and MCCEA.

4 Conclusions

In this paper, a multi-objective carrier chaotic evolutionary algorithm has been developed, and is used to produce good DNA sequences for DNA computing. The simulation results show that our algorithm is efficient to generate a set of DNA sequences with good quality. Although the multi-objective carrier chaotic evolutionary algorithm looks simple and rough, it already has many advantages, for example, it is easy to produce a set of alternate solutions and is not necessary to give weight values. The multi-objective carrier chaotic evolutionary algorithm deserves to be further investigated.

DNA sequences design problem is important not only in DNA computing but also in biology. Further research is necessary, and much work needs to be done in the future, such as developing more accurate practical model formulations, and more efficient algorithms.

References

1 Adelman LM. Molecular computation of solutions to combinatorial problems. Science, 1994, 266(5187): 102—1024

2 Lipton R. DNA solution of hard computational problems. Science, 1995, 268(28): 542—545

3 Ouyang Q, Kaplan PD, Liu S, et al. DNA solution of the maximal clique problem. Science, 1997, 278(10): 446—449

4 Braich RS, Chelyapov N, Johnson C, et al. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296 (5567): 499—502

5 Marathe A, Condon AE, Corn RM. On combinatorial DNA word design. In: Proceedings of the 5th DIMACS Workshop on DNA Based Computers. Cambridge, MA, USA, 1999, 75—89

6 Frutos AG, Thiel AJ, Condon AE, et al. DNA computing at surfaces: 4 base mismatch word design. In: Proceedings of the 3rd DIMACS Workshop on DNA Based Computers. Univ of Penns, 1997, 238—238

7 Arita M, Nishikawa A, Hagiya M, et al. Improving sequence design for DNA computing. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2000), 2000, 875—882

8 Tanaka F, Nakatsugawa M, Yamamoto M, et al. Developing support system for sequence design in DNA computing. In: Preliminary Proceedings of the 7th International Workshop on DNA-Based Computers, 2001, 340—349

9 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, Indianapolis, India, 1997, 267—272

10 Shin SY, Kim D, Lee IH, et al. Evolutionary sequence generation for reliable DNA computing. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, Honolulu, HI, USA, 2002, 79—84

11 Chui GZ, Niu YY, Wang YF, et al. A new approach based on PSO algorithm to find good computational encoding sequences. In: Pre-Proceedings of the International Conference Bio-Inspired Computing-Theory and Applications. Wuhan, 2006, 39—48

12 Athan TW and Papalambros PY. Note on weighted criteria methods for compromise solutions in multi-objective optimization. Engineering Optimization, 1996, 27(2): 155—176

13 Wetmur JG. DNA probes: applications of the principles of nucleic acid hybridization. Critical Rev Biochem Molecular Bio, 1991, 26 (3): 227—259

14 Santa L. A unified view of polymer: dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. Proc Nat Acad Sci, USA, 1998, 95: 1460—1465

15 Wang W, Zheng XD, Zhang Q, et al. The optimization of DNA encodings based on GA/SA algorithm. In: Preproceedings of the International Conference Bio-Inspired Computing-Theory and Applications. Wuhan, 2006, 49—58

16 Shin SY, Lee IH, Kim D, et al. Multi-objective evolutionary optimization of DNA sequences for reliable DNA computing. IEEE Transactions on Evolutionary Computation, 2005, 9(2): 143—158

17 Aihara K, Takabe T and Toyoda M. Chaotic neural network. Physics Letter A, 1990, 144(6): 333—340

18 Wang Z, Zhang T and Wang H. Simulated annealing algorithm of optimization based on chaotic variable. Control and Decision, 1998, 14(4): 381—384

19 Zhang T, Wang H and Wang Z. Mutative scale chaos optimization algorithm and its application. Control and Decision, 1999, 14(3): 285—288

20 Tang W. Chaotic optimization method based on power function carrier and its applications. Control and Decision, 2005, 20(9): 1043—1046

21 Coello CAC, Veldhuizen DAV and Lamont GB. Evolutionary Algorithms and Applications. London: Spring Verlag, 2005

22 Deb K. Multi-Objective Optimization Using Evolutionary Algorithm. Chichester: John Wiley & Sons, 2001

23 Barone L, While L, Hughes P, et al. Fixture-scheduling for the australian football league using a multi-Objective evolutionary algorithm. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver: Canada, July 16—21, 2006, 954—961

24 Zitzler E and Thiele L. An evolutionary algorithm for multi-objective optimization: The strength Pareto approach. Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich, Switzerland, Technical Report43, 1998