基于动态遗传算法的 DNA 序列集合设计

张 强 王 宾 张 锐 许春霞
（先进设计与智能计算省部共建教育部重点实验室（大连大学） 辽宁 大连 116622）

摘 要 DNA 编码序列的质量与数量直接影响着 DNA 计算的可靠性和规模，如何找到尽可能好的及尽可能多的 DNA 序列用于实际的应用一直是 DNA 计算的一个核心问题. 文中首先介绍了研究 DNA 编码和 DNA 序列集合对 DNA 计算的意义，并给出了 DNA 序列设计的汉明距离和反汉明距离约束条件的定义. DNA 序列集合的研究对 DNA 计算的可靠性和规模有着重要的影响，因此文中利用遗传算法和动态遗传算法来设计满足上述约束条件的 DNA 序列集合，通过对两种方法所得结果的比较，证明了动态遗传算法明显优于遗传算法. 与此同时，将文中所得到的实验结果与前人的研究成果进行比较可知，文中的结果大幅提高了 DNA 编码的上界，从而进一步缩小了 DNA 编码界的取值范围. 并且文中所给出的实验结果，对以后 DNA 编码的理论界的研究以及编码理论中关于 4 元码界的研究，提供了重要的参考值.

关键词 DNA 序列集合；遗传算法；汉明距离；反汉明距离；动态遗传算法
中图法分类号 TP384

Genetic Algorithm-Based Design for DNA Sequences Sets

ZHANG Qiang WANG Bin ZHANG Rui XU Chun-Xia
(*Key Labortory of Advanced Design and Intelligent Computing (Dalian University) of Ministry of Education, Dalian, Liaoning* 116622)

Abstract Both the quality and the quantity of DNA coding sequences immediately affect the reliability and the scale of DNA computing, and how to find more and better DNA sequences which are used in practical applications is the key problems in DNA computing. This paper firstly introduces the significance of DNA coding and DNA sequences sets for DNA computing, and then gives the definitions of the Hamming distance and reverse Hamming distance constraints. The research of DNA sequences sets has important effect on the reliability and the scale of DNA computing, so we use genetic algorithm and dynamic genetic algorithm were used to design the largest DNA sequence sets which satisfy Hamming distance and reverse Hamming distance combinatorial constraints. Comparing the results which are obtained by the two algorithms, it can be proven that the dynamic genetic algorithm is markedly better than genetic algorithm. At the same time, by comparing the obtained experimental results with the previous works, the obtained results improve the upper bounds of DNA coding, and further shorten the value range of DNA coding bounds. Furthermore, the results provide a guide for the research of theoretical bounds in DNA coding and the bounds of 4-ary in coding theory.

Keywords DNA sequence sets; genetic algorithm; Hamming distance; reverse Hamming distance; dynamic genetic algorithm

---

收稿日期：2008-04-16；最终修改稿收到日期：2008-10-17. 本课题得到国家自然科学基金（60403001，60533010，30740036，30870573）、辽宁省高校优秀人才支持计划（RC-05-07）及辽宁省高校创新团队计划（2008T004）资助. **张 强**，男，1971 年生，博士，教授，研究兴趣包括 DNA 计算、神经网络、计算机动画等. E-mail：zhangq@dlu.edu.cn. **王 宾**，男，1982 年生，硕士研究生，研究方向为 DNA 计算. **张 锐**，女，1982 年生，硕士研究生，研究方向为 DNA 计算. **许春霞**，女，1982 年生，硕士研究生，研究方向为 DNA 计算.

1 Introduction

DNA computing is a new method that uses biological molecule DNA as computing medium and biochemical reaction as computing tools. In 1994, Dr Adleman released "Molecular Computation of Solutions to Combinatorial Problems" in Science, which indicates DNA computing comes into being [1]. The DNA molecule has a high parallelism, and a great of storage capacity as a carrier of information. At the same time, the resource of DNA molecule is very rich. Along with the development of biologic technology, the DNA computing will solve more and more complex problems, especially NP problems. Finally, it may produce a new DNA computer which could bring the flying development of mathematics, computer science and other subjects.

At present, DNA computing has obtained a lot of progress in many fields, such as coding problem, realizing way, DNA computing model and arithmetic operation system[1-27].

Although the research of DNA computing has obtained a lot of enormous progress in many areas, there are a number of problems which are not solved, such as coding problem, biologic technology problem and so on. The urgence to solve these problems is that how to combine the recognition of information specific in DNA computing with varied kind of biochemical reaction factors and build a method of standard coding. So finding a strong robustness coding set to express information is the most important problem. At present, the coding problem has been investigated by more and more researchers. However, there is no best method to solve the coding problem. In the DNA coding constrains, the mostcommon constrains are distance constrains and thermodynamics constrains. The distance constrains commonly includes Hamming distance (HD), reverse Hamming distance (RH), reverse-complement Hamming distance(RC)and so on. The thermodynamics constrains commonly includes GC content, free energy ΔG, melting temperature T_M and so on.

The main problem of designing DNA sequences sets is to research the bounds of DNA coding. The bounds include two main methods: one is theoretical derivation. It can obtain the tectonic method of DNA sequences which satisfy the constraints and the approximate upper or lower bounds, such as [18-20]. The other is to use the intelligent algorithm to search the DNA sequences sets which satisfy the constraints and obtain the DNA sequences sets, such as stochastic local search algorithms[21-22], hybrid randomized neighborhoods improve stochastic local search[23], Dynamic Neighborhood Searches[24] and so on.

In the theoretical derivation, the main idea is to apply the research results of 2-component code and q-component code to the DNA coding and improve them[25], such as Sphere-Packing bound, Singleton upper bounds, Gilbert-Varshamov lower bounds, Plotkin lower bounds and so on. There are some introductions in the [18] and [19], and some corresponding derivation. Applying these results, they reduce the range of sets value. In [19], the authors deeply researched the theoretical bounds which satisfy the HD. In [18], the authors deeply researched the theoretical bounds which respectively satisfy the HD, HD and RC, HD and HR, and gave the relation of them. In [25], the authors researched the GC content and RC and GC content constraints. In [26], the authors used the method which combines linear construction with stochastic local search algorithm. They improved the some lower bounds which satisfy the GC content and RC and GC content constraints.

In the research of intelligent algorithm, the main idea is to use intelligent algorithm to search DNA sequences which satisfy constraints. In the single constraint, this method usually is used to improve the lower bounds. In the combinatorial constraints, this method can improve the upper and lower bounds. Because the theoretical research is hard to find the relation between the combinatorial constraints, such as the relation between GC content and other distance constraints. In [21], the author used the stochastic local search algorithm to improve the lower bounds which satisfy the HD and RC combinatorial constraints. The results were compared with the theoretical value. At the same time, they also improved the bounds which satisfy the HD and RC constraints and obtain the approximate bounds which satisfy the RC and GC content constraints. In [23], the authors improved the stochastic local search algorithm and the results which are from the [21]. In [24], the authors used the dynamic neighborhood search to improve the lower bounds which satisfy the RC and GC content constraints.

In this paper, we mainly research the design of DNA sequences sets which satisfy combinatorial constraints. We firstly use the genetic algorithm to obtain a great deal of sets which satisfy the combi-

natorial constraints. then, we use the dynamic genetic algorithm to obtain a great deal of sets which satisfy the combinatorial constraints. Comparing with the results which are from the genetic algorithm, the results which are from dynamic genetic algorithm is better than them and proof that the dynamic genetic algorithm is better than genetic algorithm in the designing DNA sequences sets.

2 Constraints[2,11-13]

Garzon firstly proposed the definition of coding problem in DNA computing. The definition is as follow: in the alphabet $\Sigma=\{A,G,C,T\}$, it exists a set S with the length of n. The size of S, $|S|=4^n$. A subset of S, $C\subseteq S$ and let x_i, x_j any two codes in the C satisfy

$$\tau(x_i,x_j)\geq k \tag{1}$$

where k is positive integer, τ is the criterion of estimating the quality of coding, such as Hamming distance, shift distance and so on. Obviously, the main index is the quantity and the quality of coding in the problem of coding. The larger quantity of coding is, the larger application scale of solving problem is; on the other hand, the better quality of coding is, the more reliability of DNA computing is. However, the quantity is in contradiction with the quality of coding in the practical problems. The better quality of coding is, the more number of constraints, and then it must lead to decrease the number of DNA sequences which satisfy constraints. In this paper, we use Hamming distance, reverse Hamming distance and reverse complement Hamming distance under the Hamming criterion and GC content constraints. n is the length of DNA sequences, d is the Hamming distance, reverse Hamming distance and reverse complement Hamming distance. The follows are their definitions.

2.1 Hamming Distance Constraint

Hamming constraint: the Hamming distance between x_i and x_j should not be less than certain parameter d, i. e. $H(x_i,x_j)\geq d$.

$$f'_{\text{Hmming}}(i)=\min_{1\leq j<m,\ j\neq i}\{H(x_i,x_j)\} \tag{2}$$

Where $f'_{\text{Hamming}}(i)$ indicates the Hamming evaluation function of the ith individual in evolutionary population. In this paper, we let $A_4^{HD}(n,d)$ denote the maximum size of a Hamming distance code, with parameters n,d defined as for codes.

2.2 Reverse Hamming Constraint

Reverse Hamming constraint: the Hamming distance between x_i and x_j^R should not be less than a certain parameter d, i. e. $H(x_i,x_j^R)\geq d$. x_j^R denotes the reverse of x_i.

$$f'_{\text{Reverse}}(i)=\min_{1\leq j<m}\{H(x_i,x_j^R)\} \tag{3}$$

Let $A_4^{RH}(n,d)$ denote the maximum size of a reverse Hamming distance code, with parameters n,d defined as for codes.

2.3 Fitness Function

The optimization problem is defined by the problem of maximum value and we use average weight to deal with the function of evaluation of each constraint.

$$f_j\in\{f_{\text{Hamming}}(i),\ f_{\text{Reverse-comple}}(i)\}$$

$$f(i)=\sum_{j=1}^{m}\omega_j f_j(i) \tag{4}$$

where ω_j is the weight of the each constraint, m is the number of constraints. In this paper, $\omega_j=1$ and $m=2$, when the constraints are Hamming distance and reverse Hamming distance combinatorial constraints. $\omega_j=1$ and $m=2$, when the constraint is Hamming distance, reverse Hamming distance, reverse-complement Hamming distance.

3 Algorithm Design

In this paper, we use the genetic algorithm and dynamic genetic algorithm to design DNA sequences sets which satisfy combinatorial constraints. Genetic Algorithm (GA) is stochastic search algorithm based on nature selection and genetic mechanism. GA could solve a number of problems, because the process of nature evolution is a process of learning and optimizing. The main idea of this algorithm is that: the process of nature evolution(from simple to complex, from low class to upper class)is natural and parallel; the intention is to adapt to environment. The biological populations began to evolve by survival of the fittest and genetic variation. The evolution of biology is carried out by propagate, mutation, competition and selection.

The genetic algorithm simulates the structure of survival of the fittest and the theory of genetic mutation. It keeps the structure in the iterative process, when it finds the better structure. As a stochastic optimized and search algorithm, genetic algorithm has a lot of advantages, as follows: it is not a blindfold search and exhaustive algorithm, it is an algorithm which is based on objective function; it only needs to use the objective information and do not need the gradient, continuity, convexity and so on. So it could be used to optimize the large scale, highly nonlinear discontinuity and peaky functions and the function of non-analytic expression and has very strongly generality; the operational object of genetic algorithm is a group

object, not a single object. It has many search trajectories, so it has implicit parallelism.

3.1 Genetic Operators

In genetic algorithm, it includes three operators: selection operator, crossover operator and mutation operator.

(1) Selection operator

The selection strategy plays a very important role on the ability of algorithm. Generally, the selection strategy not only affects the ability of algorithm, but also the results. The selection strategy usually uses fitness-proportion strategy, namely roulette wheel selection. In the strategy, every individual is proportional to its fitness. The size of the population is n, the fitness of individual i is f_i, then the probability is P_{si}.

$$P_{si} = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

P_{si} reflects proportion of individual fitness of in the whole population fitness. Based on the size of individual fitness, the larger fitness, the larger P_{si}. So the selective and genetic change of individual becomes the larger.

(2) Crossover operator

In the genetic algorithm, it is very important for the algorithm. On the one hand, it keeps the good characteristic of original population to some extent. On the other hand, it makes the algorithm to search new population and the new population have diversity. Crossover operator usually uses One-point Crossover, Two-points Crossover, Uniform Crossover and so on. In this paper, we use One-point Crossover, namely simple Crossover. Its operations are as follows: stochastically setting a crossover point in the single population; when doing crossover, exchanging the pre and post parts of this point and generate two new populations. When the length of population is n, there maybe $n-1$ crossover points.

(3) Mutation operator

The main content of mutation operator is to vary value in some sits of one population. Generally, the main processes of mutation operator are as follows:

(i) Stochastically deciding the sit of population in the whole populations.

(ii) Using P_m to set the mutation operator.

In the genetic algorithm, crossover operator is the main operator because of its Global searching ability. The mutation operator is the assistant operator because of its local searching ability. Genetic algorithm has Global and local searching ability by these two operators combining or competing each other. The combining is that when the evolution traps in a searching hyperplane and can not cast off by crossover operator, using the mutation operator can help it cast off; The competing is that when the crossover operator has obtained the expected populations, the mutation operator maybe destroy these populations. Using the mutation operator could make genetic algorithm have local searching ability and diversity populations.

3.2 Designing DNA Sequences Sets Based on Genetic Algorithm

The main idea of genetic algorithm is that: firstly stochastically generating new DNA sequences, controlling the aspect of evolution by controlling the fitness and selecting DNA sequences which satisfy HD and RH combinatorial constraints; then using the genetic operators to generate new DNA sequences which satisfy HD and RH combinatorial constraints; finally obtaining the DNA sequences sets. Fig. 1 is the flowchart of algorithm.

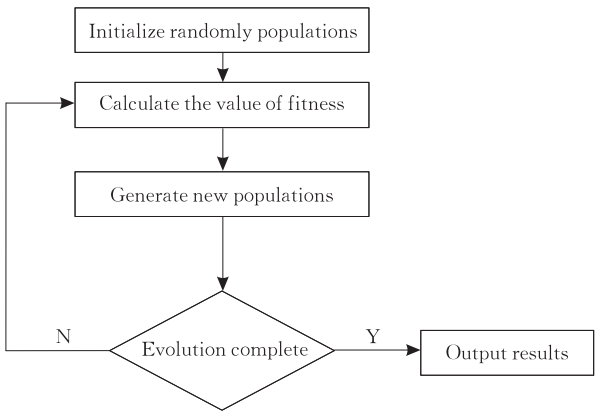

Fig. 1 Flowchart of algorithm

The steps of genetic algorithm are as follows:

Step 1. Set parameter, and initialize population randomly.

Step 2. Calculate the fitness value of every individual in the populations and selecting the DNA sequences which satisfy the combinatorial constraints.

Step 3. Generate next population by select、crossover and mutate. If the generation is less than 3000, go to step2; if not then go to step4.

Step 4. end and output results.

3.3 Dynamic Genetic Algorithm

The dynamic genetic algorithm is to improve the genetic algorithm. To obtain the more DNA sequences which satisfy the constraints, we control

the evolution by controlling the fitness. The improved areas are as follows:

(1) When calculating the value of fitness, we use the dynamic manner to evolve the populations.

(2) In the selection process, we use best-saved strategy.

(3) In the mutation process, we adjust the probability of mutate operator with dynamic method.

The main process is that: Generating DNA sequences by the dynamic genetic algorithm, and controlling the direction of evolution by controlling the value of fitness function, and selecting the sequences which satisfy the Hamming distance, reverse complement Hamming distance constraints from these sequences, and then generating new DNA sequences by select, crossover and mutate operator, last obtaining the DNA sequence sets. Fig. 2 is the flowchart of algorithm.

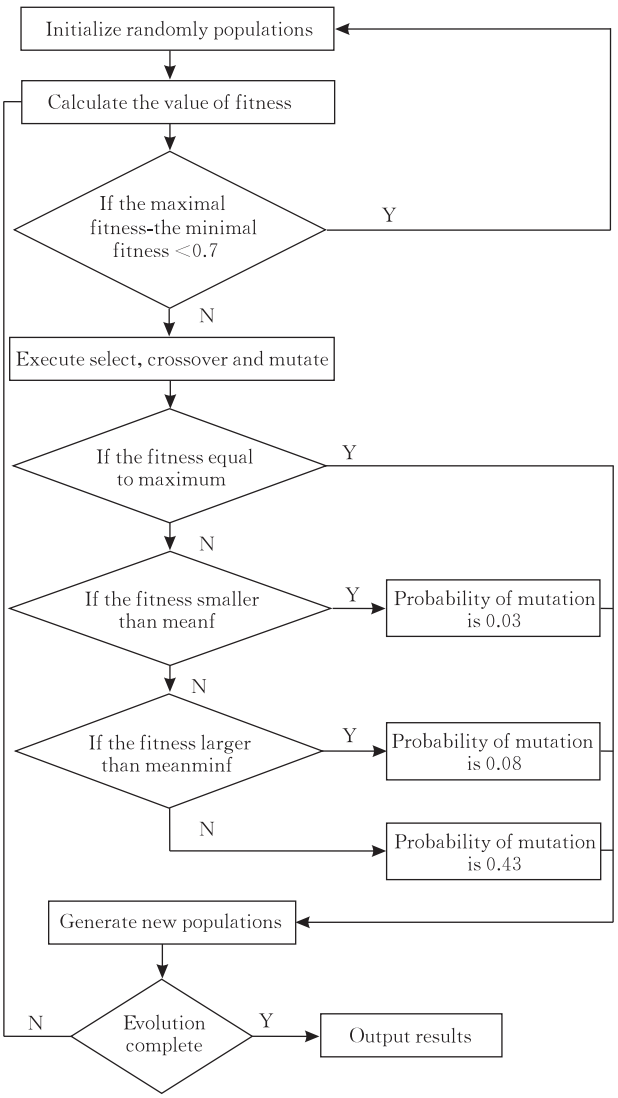

Fig. 2　The flowchart of algorithm

The steps of solving the sequence design by Dynamic Genetic algorithm are as follows:

Step 1. Set parameter, and initialize population randomly. 1.

Step 2. Calculate the fitness value of every individual in the populations. If the maximal fitness—the minimal fitness <0.7, go to Step 1. 1.

Step 3. Generate next population by select、crossover and mutate. In the selection process, if one of fitness is the maximal value, then save this population to the next generation (namely best-saved strategy). In the mutation process, the meanf denotes the average fitness of populations. The meanminf denotes the average of the meanf and minimum fitness. If one of fitness is larger than meanf, its probability of mutation is 0.03. And if it is smaller than meanf, but the fitness is larger than meanminf, its probability of mutation is 0.05. Else, its probability of mutation is 0.4. If the generation is less than 500, go to Step 2; if not then go to Step 4. 1.

Step 4. End and output results.

4　Computational Experiments

The parameters of GA used in our example are: the size of population is 40. The probability of crossover is 0.7. The probability of mutate which is initialized is 0.03. The generation is 3000. The parameters of dynamic GA used in our example are: the size of population is 40. The probability of crossover is 0.7. The probability of mutate which is initialized is 0.03. The generation is 500. Table 1 which is obtained by GA is the bounds of DNA sequence sets that satisfy the Hamming and reverse Hamming constraints as a function of word length n and distance d. We do five experiments for every value and use the maximum. Table 2 which is obtained by dynamic GA is the bounds of DNA sequence sets that satisfy the Hamming and reverse Hamming constraints as a function of word length n and distance d. The value is the minimum generation of one experiment in the bracket. We do five experiments for every value.

It is noted that because of continuously increasing n the value of $A^R(n,d)$ rapidly increase. So the computational complexity rapidly increases. Table 3 is the best previously known bounds of DNA sequence sets that satisfy the Hamming and reverse Hamming constraints as a function of word length n and distance d[18].

Table 1 Results of $A_4^R(n,d)$ by GA

n	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$
4	27	**6**	**2**	-	-	-	-
5	81	**13**	**4**	0	-	-	-
6	225	**41**	**12**	**3**	**2**	-	-
7	945	**141**	**28**	**7**	**3**	0	-
8	1495	**410**	66	**18**	**6**	**2**	**2**

Table 2 Results of $A^R(n,d)$ by Dynamic GA

n	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$
4	**32**!(1857)	**6**!(2)	**2**!(1)	-	-	-	-
5	94!(358)	**20**!(158)	**4**!(1)	0	-	-	-
6	316!(29)	**49**!(122)	**14**!(357)	**4**!(1)	**2**!(1)	-	-
7	1109!(49)	**147**!(6)	**35**!(486)	**9**!(256)	**3**!(1)	0	-
8	3501!(4)	**453**!(10)	95!(1060)	**23**!(63)	**8**!(8)	**2**!(1)	**2**!(1)

Table 3 The Best Previously Known Results of $A^R(n,d)$[18]

n	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$
4	32	2−8	2	-	-	-	-
5	120−128	4−26	2−8	0	-	-	-
6	512	12−107	4−107	2−5	2	-	-
7	$2016-2^{11}$	33−372	8−372	2−34	2−4	0	-
8	2^{13}	160−1310	128−1310	4−118	2−118	2−3	2

In the Table 1, entries in bold face match or exceed the previously known lower bounds. There do not exist values in the blank which have '-'. The others are lower the previously known lower bounds. In the Table 2, entries in bold face match or exceed the previously known lower bounds. There match or exceed the values of Table 1 in the blank which have '!'.

According to Table 1, we can obtain conclusions that the GA can be used to design DNA sequences sets which satisfy combinatorial constraints and obtain better results. At the same time, comparing with Table 2, we can see that the dynamic GA is better than the GA for the designing DNA sequences sets which satisfy combinatorial constraints. So far, there are no accurate values about $A^R(n,d)$, so the results of $A^R(n,d)$ obtaind here improve the lower bounds of DNA coding, and further shorten the value range of DNA coding bounds. Furthermore, the results provide important referenced values for the research of theoretical bounds in DNA coding and the bounds of 4-ary in coding theory.

5 Conclusion

In this paper, we use the genetic algorithm and dynamic genetic algorithm to design the DNA sequences sets which satisfy the HD and RH combinatorial constraints, namely $A^R(n,d)$. It is shown that the dynamic genetic algorithm is better than the genetic algorithm. When n is increasing, if we increasing the number of population, the number of generations maybe increase. However, the computational complexity and computing time will largely increase. It must be noticed that though there have been a lot of researchs on the DNA computing since the DNA computing generating, the theoretic research is deficient, especially the research on 4 component code, namely the bounds of DNA sequences sets which satisfy the combinatorial constraints. For this reason, we need to do a great of deeply research on the designing DNA sequences sets which satisfy the combinatorial constraints.

To the best of our knowledge, there is no result on the theoretic accurate value of DNA sequences sets which satisfy the combinatorial constraints. So the DNA sequences sets which we obtain will provide important referenced values for the bound of $A^R(n,d)$ and help to obtain the accurate value. Furthermore, the results provide important referenced values for the research of theoretical bounds in DNA coding and the bounds of 4-ary in coding theory.

Moreover, there is no structure method which satisfies the combinatorial constraints in the theoretic research on the designing DNA sequences sets, so it is very hard to apply them to DNA computing. According to our algorithm, we can obtain the DNA sequences which satisfy the combinatorial constraints and use them into DNA computing to decrease the he emergence of false negative and false positive, and improve the efficiency and reliability of DNA computing. At the same time, using these results can decrease the redundancy of DNA data storage.

There are many problems which need to be improved, such as the time complexity of algorithm, the stability of algorithm and so on. For these reasons, we will further improve our algorithm and increase more practical constraints. At the same time, we will study the theoretic bounds of DNA sequences sets which satisfy combinatorial constraints.

References

[1] Adleman L M. Molecular computation of solution to combinatorial problems. Science, 1994, 266: 1021-1024

[2] Baum E B. DNA sequences useful for computation//Proceedings of the 2nd DIMACS Workshop on DNA Based Computers. Princeton University, 1996: 122-127

[3] Deaton R, Murphy R C, Rose J A, Garzon M, Franceschetti D R, Stevens Jr S E. A DNA based implementation of an evolutionary search for good encodings for DNA computation//Proceedings of the 1997 IEEE International Conference on Evolutionary Computation. Piscataway, NJ, 1997: 267-272

[4] Deaton R, Franceschetti D R, Garzon M, Rose J A, Murphy R C, Stevens S. E. Information transfer through hybridization reaction in DNA based computing//Proceedings of the

Second Annual Conference on Genetic Programming. Stanford University, San Francisco, CA, USA, 1997: 463-471

[5] Garzon M, Neathery R, Deaton R C, Murphy D R, Franceschetti S E, Stevens Jr. A new metric for DNA computing//Proceedings of the 2nd Annual Genetic Programming Conference GP-97. Stanford University, 1997: 472-487

[6] Wood D H. Applying error correcting codes to DNA computing//Proceedings of the 4th DIMACS workshop on DNA based computers. Pennsylvania, PA, 1998: 109-110

[7] Hartemink A, Gifford D, Khodor J. Automated constraint-based nucleotide sequence selection for DNA computation//Proceedings of the 4th DIMACS Workshop on DNA Based Computers. Pennsylvania, PA, 1998: 287-297

[8] Hartemink A, Gifford D K. Thermodynamics simulation of deoxyoligonucleotide hybridize for DNA computation//Proceedings of the 3rd DIMACS Meeting on DNA Based Computers. Pennsylvania, PA, 1997: 23-25

[9] Shin S Y, Kim D M, Lee I H, Zhang B T. Evolutionary sequence generation for reliable DNA computing//Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu, Hawaii, USA, 2002: 79-84

[10] Lipton R. DNA solution of hard computation problems. Science, 1995, 268: 542-545

[11] Ouyang Q, Kaplan P D, Liu S, Libchaber A. DNA solution of the maximal clique problem. Science, 1997, 78(2): 446-449

[12] Liu Q, Wang L, Frutos A G, Condon A E, Corn R M, Smith L M. DNA computing on surfaces. Nature, 2000, 403: 175-179

[13] Smith L M, Corn R M, Condon A E, Lagally M G, Frutos A G, Liu Q, Thiel A J. A surface-based approach to DNA computation. Journal of Computational Biology, 1998, 5(2): 255-267

[14] Ravinderjit S B, Nickolas C, Cliff J, Paul W K R, Adleman L M. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296: 499-502

[15] Guarnieri F, Fiss M, Bancroft C. Making DNA add. Science, 1996, 273: 220-223

[16] Berard Y, Allen P, Cheng S L. DNA implementation of addition in which the input strands are separated from the operater strands. Biosystem, 1999, 52: 165-174.

[17] Hug H, Schuler R. DNA-based parallel computation of simple arithmetic//Proceedings of the 7th international Workshop on DNA-Based Computers, DNA 2001. Tampa, USA, 2001: 159-166

[18] Marathe A, Condon A, Corn R. On combinatorial DNA word design. Journal of Computational Biology, 2001, 8(3): 201-220

[19] Galina T B, Brouwer A E, Kapralov S N, Patric R J. Error-correcting codes over an alphabet of four elements. Designs, Codes and Cryptography, 2001, 23(3): 333-342

[20] Paolo B, Anna L, Vincenzo M, Victor M. Superposition based on Watson—Crick-Like complementarity. Theory of Computing Systems, 2006, 39(4): 503-524

[21] Tulpan D, Hoos H, Condon A. Stochastic local search algorithms for DNA word design//Proceedings of the 8th International Workshop on DNA-Based Computers. Sapporo, Japan, 2002, 2568: 229-241

[22] Chee Y M, Ling S. Improved lower bounds for constant GC-content DNA codes. IEEE Transactions on Information Theory, 2008, 54(1): 391-394

[23] Tulpan D, Hoos H. Hybrid randomized neighborhoods improve stochastic local search for DNA code design//Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence. Halifax, Canda, 2003, 2671: 418-433

[24] Suguru K, Hirotaka O, Kunihiko S, Masafumi Y. DNA sequence design by dynamic neighborhood searches. Seoul, Korea, 2006, 4287: 157-171

[25] King O D. Bounds for DNA codes with constant GC-content. Electronic Journal of Combinatorics, 2003, 10(1): 1-13

[26] Gaborit P, King O D. Linear constructions for DNA codes. Theoretical Computer Science, 2005, 334(1-3): 99-113

[27] Wang W, Zheng X, Zhang Q, Xu J. The optimization of DNA encodings based on GA/SA algorithms. Progress in Natural Science, 2007, 17(6): 739-744

ZHANG Qiang, born in 1971, Ph.D., professor. His cuurent research interests include DNA computing, neural networks and computer animation.

WANG Bin, born in 1982, graduate student. His cuurent research interest is DNA computing.

ZHANG Rui, born in 1982, graduate student. Her cuurent research interest is DNA computing.

XU Chun-Xia, born in 1982, graduate student. Her cuurent research interest is DNA computing.