0-1 Integer Programming Based on DNA Tetrahedral Probe

Jing Yang[1], Xinmu Yang[1], Zhixiang Yin[1](✉), Zhang Qiang[1,2],
and Jianzhong Cui[1]

[1] School of Mathematics and Big Data,
Anhui University of Science and Technology, Huainan, Anhui, China
jyangh82@163.com, zxyin66@163.com
[2] School of Computer, Dalian University of Technology,
Dalian, Liaoning, China

Abstract. It is difficult to find an effective algorithm for solving NP complete problems such as integer programming. The nanostructure constructed by DNA origami combines huge parallelism and massive storage capacity of DNA computing. In the calculation process, it can effectively avoid the number of experimental operations required by other DNA computing models. It greatly reduces the time consumption and the rate of misinterpretation, thus providing an effective way to efficiently solve integer programming. DNA tetrahedron is a nanostructure constructed by origami. It has stable structure, good toughness and compression resistance, simple production process, high yield, rich functional modification sites, good biocompatibility, but also resistance to a variety of specific or non-specific nuclease. Therefore it can reduce the misinterpretation rate of biochemical reactions using DNA tetrahedron and DNA single strand to construct probes, finding the true solution according to the constraint condition. And then it can improve the computational efficiency of the model.

Keywords: DNA tetrahedron · 0-1 integer programming · DNA origami · DNA computing

1 Introduction

DNA nanotechnology has been developing rapidly in the past 30 years [1, 2]. There are two main ways to construct self-assembled DNA nanostructures, including tile self-assembly developed by Seeman's team [3, 4] and DNA origami invented by Rothemund [5]. In 1980s, Seeman [6] put forward "structural DNA nanotechnology". Researchers then constructed different modules, such as DX (double-crossover) module [7–11], TX (triple-crossover) module [12], cross module [13] and symmetry module [14], and assembled various graphical structures (two-dimensional arrays, square meshes, polyhedrons, etc.) [15]. In 2006, Rothemund [5] proposed a new method of DNA self-assembly, DNA origami. By DNA origami, Rothemund obtained the intricate two-dimensional structures of triangles, squares, rectangles, pentagrams and smiling faces. Rothemund's research laid the foundation for the precise assembly of nanostructures, and then more complex and diverse nanostructures [16, 17] were

H. Peng et al. (Eds.): ISICA 2018, CCIS 986, pp. 91–100, 2019.
https://doi.org/10.1007/978-981-13-6473-0_9

constantly designed and assembled, providing more sophisticated templates for the in-depth study of nanodevices. In 2007, Shih et al. [18] folded M13 as a scaffold stand to form 410 nm hexahelical nanotubes using DNA origami and site-specific design. This is the first time that researchers have used DNA origami to create a three-dimensional structure, which opens the prelude to the assembly of three-dimensional nano-DNA structures. Subsequently, the researchers designed and obtained many complex and exquisite three-dimensional DNA self-assembly structures.

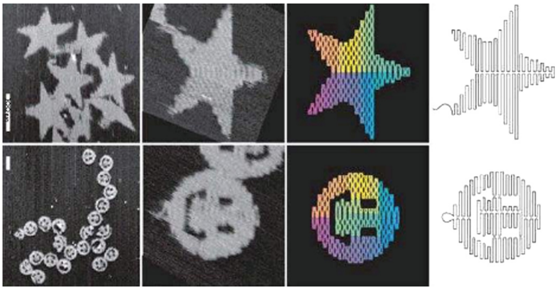


Fig. 1. Smiles and stars folded in DNA origami

In 2008, Andersen et al. [19] successfully designed the school badge of dolphin structure of Aarhus University using DNA origami. By adjusting the number of Crossover at the tail to control the movement of the dolphin tail, the shapes of different shapes were obtained. This is the first asymmetric figure with holes. The next year, they designed a 3D box with M13mp18 as the scaffolding stand, and switched the lid of the box through the stand replacement reaction. Three months later, the lab's Dietz et al. devised a variety of twist patterns [20], with angles accurate to about 3 degrees. Soon after the separation, Ke et al. [21] designed a hollow tetrahedral structure. Subsequently, Han et al. [22] pushed the DNA origami to a climax in 2013. They used a nearly vertical DNA strand composed of a cross structure, and successfully obtained the structure of 21 bp, 42 bp, 63 bp network structure. Then by adjusting the scaffolding assembly mode, the node is slightly twisted to form a gridiron-like structure. The three-dimensional spherical structure and spiral structure of DNA were successfully designed to overcome the problem of charge mismatch of DNA molecules. Han et al. have greatly promoted the preparation of three-dimensional DNA nanostructures. The cavity of three-dimensional DNA nanostructures can be used to carry some drug molecules to achieve targeted therapy of diseases. In recent years, the research on DNA tetrahedron has attracted more and more scholars' attention. DNA tetrahedron structure is a three-dimensional DNA nanostructure with tetrahedron shape, which is composed by DNA origami, ingenious DNA sequence design, complementary pairing principle and automatic hybridization of each strand (Fig. 1).

The tetrahedron is closed by 4 triangular planes, and each side of the triangle is about 54 nm. It is hybridized by multiple staple stands and a scaffold of DNA, rather than each other, so the response requires relatively low accuracy in the measurement relationship. It can significantly improve the success rate and yield of the assembly.

DNA tetrahedron is a kind of nanostructure. It has stable structure, good toughness and compressive properties, simple manufacturing process, high yield, rich functional modification sites, good biocompatibility [23]. It can also resist a variety of specific or non-specific nucleases, and has good application potential in molecular diagnosis, bioimaging, molecular delivery and drug targeted therapy.

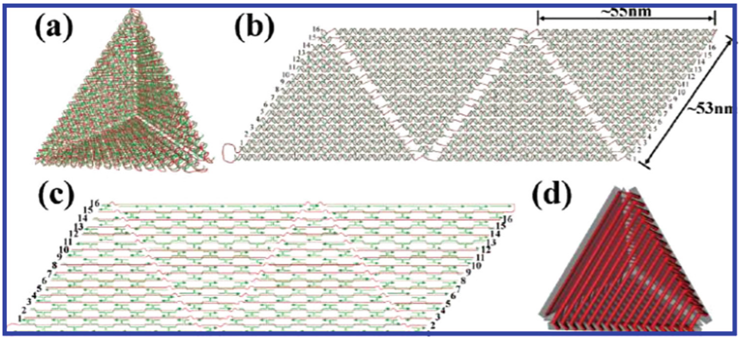


Fig. 2. The design of DNA tetrahedron by Ke et al. [21]

At present, there are many software packages for the design of DNA nanostructures [24], such as Uniquimer 3D [25] and NANEV [26]. The development and application of these software packages make it easier to design DNA nonmaterial including tetrahedral structures. DNA tetrahedrons suitable for integer programming will be constructed by DNA origami technology, and the computational complexity of integer programming will be reduced by combining the advantages of nanostructures in this paper (Fig. 2).

2 0-1 Integer Programming

Integer programming is an independent branch [27] formed by R.E. in 1958 after the cutting plane method was proposed. Generally, it is consistent with the combinatorial optimization problem, which is to find the best solution satisfying certain constraints among a limited number of alternatives. Such as backpack (or loading) problem, fixed cost problem, harmony exploration team problem (combinatorial pairing problem), effective exploration team problem (combinatorial coverage problem), traveling salesman problem, vehicle routing problem and so on. It can be transformed into integer programming problem to solve. Its method mainly includes branch and bound method, cutting plane method and exhaustive method. In addition, it also has wide applications in computer design, system reliability, coding and economic analysis.

Integer programming is an optimization problem with integer variables. That is, the optimization problem of maximizing or minimizing a multivariate function with all or part variables as integers under the constraints of a set of equations or inequalities. Integer programming is an important part of mathematical programming, and it has applications in many aspects of life. Nonlinear integer programming can be divided into linear part and integer part, so integer programming is often regarded as a special part of linear programming. In linear programming problems, some of the optimal solutions may be fractions or decimal. But for some specific problems, it is often required that the answer must be integers. In order to satisfy the integer requirement, it seems at first that only rounding off the non-integer solution already obtained is enough. In fact, the integer is not necessarily the feasible solution and the optimal solution, so there should be a special method to solve the integer programming. In integer programming, if all variables are restricted to integers, it is called pure integer programming. If only one variable is restricted to integers, it is called mixed integer programming. A special case of integer programming is the 0-1 programming, whose variables are limited to 0 or 1.

The general integer linear programming problem models are as follows:

$$\max(\min)z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le (=, \ge)b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \le (=, \ge)b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \le (=, \ge)b_m \\ x_1, x_2, \cdots, x_n = 0, 1, a_{ij}, b_j \text{ is integer} \\ i = 1, 2, \cdots, m.j = 1, 2, \cdots, n \end{cases}$$

0-1 programming plays an important role in integer programming. On the one hand, many practical problems, such as assignment, location and delivery, can be attributed to such programming. On the other hand, any integer programming with bounded variables can be transformed into 0-1 programming [28]. Many nonlinear programming problems can also be expressed as integer programming problems by 0-1 programming method, so many scholars are devoted to this direction of research.

When the variable x_i takes only 0 and 1, and b_j takes the non negative integer, that is 0-1 integer programming, as follows:

$$\max(\min)z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le (=, \ge)b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \le (=, \ge)b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \le (=, \ge)b_m \\ x_1, x_2, \cdots, x_n = 0, 1, b_j \text{ is } \mathit{non\ negative\ integer} \\ i = 1, 2, \cdots, m.j = 1, 2, \cdots, n \end{cases}$$

When the coefficient a_{ij} only takes 0 and 1, the variable x_i takes only 0 and 1, and b_j takes the non negative integer, that is 0-1 integer programming, as follows:

$$\max(\min)z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq (=, \geq)b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq (=, \geq)b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq (=, \geq)b_m \\ a_{ij}, x_j = 0, 1, b_j \text{ is nonnegative integer} \\ i = 1, 2, \cdots, m.j = 1, 2, \cdots, n \end{cases}$$

Since Adleman solved the problem of 7-vertex directed Hamilton path with DNA computing method in 1994 [29], DNA computing has become a hot topic for many scientists. Many scholars have tried to solve the integer programming problem with this method. The first breakthrough was the DNA computing model proposed by Yin in 2003 to solve the general 0-1 integer programming problem [30]. In the same year, Zhang Fengyue applied the surface DNA computing model to solve the 0-1 integer programming problem [31]. Yin et al. also applied DNA computing to 0-1 integer programming problem. He used DNA computing to solve a special 0-1 integer programming problem, that is, the generalization of assignment problem [32]. On the basis of reference [32], Wang gave an algorithm to solve the optimal solution of a special integer programming problem based on DNA computing [33]. Zhou et al. further extended the application of DNA computing model in 0-1 integer programming, and proposed 0-1 integer programming model with negative coefficients calculated by DNA [34]. Yang et al. gave the DNA calculation model [35] based on the three helix structure. Zhang et al. gave a self assembling DNA computing model [36] for the 0-1 integer programming problem and so on [37–39]. Integer programming theory and algorithm research is blending with other subjects of mathematical programming, but many algorithms cannot be applied to solve practical problems in society. DNA tetrahedron has the advantages of nanostructure and biochemical reaction. Therefore, it is meaningful to study the application of DNA tetrahedron structure in integer programming.

It is just beginning that using of DNA origami computing model is to solve integer programming problem. At present, the existing research results are basically around the establishment of 0-1 integer programming problem DNA computing model. Therefore, there are worthy of further study that how to construct the DNA origami computing model of 0-1 integer programming problem, how to realize the self-assembly computing model of 0-1 integer programming problem biologically, how to establish the self-assembly computing model of general integer programming problem and how to realize these problems.

3 0-1 Integer Programming Model Based on DNA Tetrahedron

3.1 Basic Algorithm

Step 1: Generate all possible combinations of variable 0 or 1 for given problem.

Step 2: Eliminate unfeasible solutions using every constraint inequality, and preserve feasible solutions.

Step 3: Generating residual solutions.

Step 4: Continue step 2,3 with the constraint inequality order, eliminate all unfeasible solutions and preserve residual solutions.

Step 5: Compare the value for object function corresponding with every feasible solution to obtain the optimum solutions.

3.2 Biological Algorithm

Step 1. We use DNA encoder to generate DNA single strand representing 0 and 1 values of n variables. There are 2n kinds of DNA short strands. Only n kinds of short strands are involved in reaction. The combination of 2n kinds of short strands should be $\underbrace{C_2^1 C_2^1 \cdots C_2^1}_{n} = 2^n$.

Step 2. In a certain constraint condition, if the variable is expressed as 1, it will not take 0 again. Similarly, if it is taken as 0, it will not take 1. Therefore, when constructing a data pool, only n kinds of DNA short strands are put into the test tube, and ligases are added to generate 2^n kinds of DNA long strands as the data pool.

Step 3. DNA tetrahedron is constructed by DNA origami. Three vertices of tetrahedron are modified and fixed on the chip. DNA single strands representing complementary strands with different constraints are constructed. DNA single strands are fixed on the remaining vertices of tetrahedron to construct DNA tetrahedron probes.

Step 4. The DNA tetrahedron probe is used to extract the DNA long stand satisfying the first constraint condition. The data pool at this time satisfies the first constraint.

Step 5. For other constraints, repeat steps (4) to exclude all solutions which do not satisfy the conditions. Finally, DNA strand satisfying all the conditions is obtained.

Step 6. Use PCR- amplification technology to read and interpret.

3.3 Case Analysis

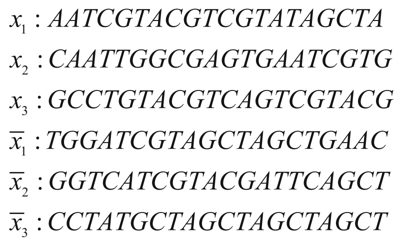
To illustrate this algorithm, we use a simple 0-1 integer programming problem to demonstrate the biological algorithm. For a general 0-1 integer programming problem, $\mathrm{a_{ij}x = x_{j1} + x_{j2} + \cdots + x_{jk}, k = a_{ij}, i = 1, 2, \cdots, m, j = 1, 2, \cdots, n}$, can be classified as a simple 0-1 programming [28].

$$\min Z = 2x_1 - 3x_2 + 2x_3$$
$$\text{s.t.} \begin{cases} x_1 + x_2 - x_3 \geq 1 \\ x_1 + x_3 \leq 1 \\ x_2 + x_3 \leq 1 \\ x_1, x_2, x_3 = 0, 1 \end{cases}$$

In order to solve the 0-1 programming problem of the above formula, the 0-1 programming is reduced to a simple 0-1 programming. $x_1 + x_2 - x_3 \geq 1$ is represented as $x_1 + x_2 - (1 - \bar{x}_3) \geq 1$. $x_1 + x_2 + \bar{x}_3 \geq 2$ is reorganized. In this way, the above 0-1 integer programming becomes a simple 0-1 programming.

$$\min Z = 2x_1 - 3x_2 + 2x_3$$
$$\text{s.t.} \begin{cases} x_1 + x_2 + \bar{x}_3 \geq 2 \\ x_1 + x_3 \leq 1 \\ x_2 + x_3 \leq 1 \\ x_1, x_2, x_3, \bar{x}_1, \bar{x}_2, \bar{x}_3 = 0, 1 \end{cases}$$

Step 1: Construct 6 kinds of oligonucleotide fragments to express $x_1, x_2, x_3, \bar{x}_1, \bar{x}_2, \bar{x}_3$ If x_i takes $1, \bar{x}_i$ takes $0, i = 1, 2, 3$ For these 6 variables, there will be no $\bar{x}_i$ when x_i occurs in constraints. Therefore, ligases can be used to generate arbitrary combinations of 2^3 = 8 kinds of DNA single strands without repetitive variables, and these eight single strands can be put into a test tube to form a data pool, (x_1, x_2, x_3),$(x_1, x_2, \bar{x}_3)$ $(x_1, \bar{x}_2, x_3)$,$(\bar{x}_1, x_2, x_3)$,$(\bar{x}_1, \bar{x}_2, x_3)$,$(\bar{x}_1, x_2, \bar{x}_3)$,$(x_1, \bar{x}_2, \bar{x}_3)$,$(\bar{x}_1, \bar{x}_2, \bar{x}_3)$ (Fig. 3).


x_1 : *AATCGTACGTCGTATAGCTA*

x_2 : *CAATTGGCGAGTGAATCGTG*

x_3 : *GCCTGTACGTCAGTCGTACG*

$\bar{x}_1$: *TGGATCGTAGCTAGCTGAAC*

$\bar{x}_2$: *GGTCATCGTACGATTCAGCT*

$\bar{x}_3$: *CCTATGCTAGCTAGCTAGCT*

Fig. 3. Encoding of variable $x_1, x_2, x_3, \bar{x}_1, \bar{x}_2, \bar{x}_3$

Step 2: DNA nanotetrahedron was constructed by origami, and the three vertices of the DNA tetrahedron were modified and fixed on the chip. The first constraint stand is encoded to construct DNA single stand. Then it is fixed on the vertex of the tetrahedron. Constitute DNA tetrahedral probe (as follow Fig. 4). The first constraint is $x_1 + x_2 + \bar{x}_3 \geq 2$. That is to say, at least two of $x_1, x_2, \bar{x}_3$ will take 1 to establish. So the combination $(x_1, x_2, \bar{x}_3)$ of constraint conditions is (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1). In order to extract the DNA strand satisfying the constraint condition 1, we need to construct these four probes. The combination (x_1, x_3) satisfying the second constraint condition is (0, 0), (0, 1), and (1, 0). The combination (x_2, x_3) satisfying the third constraint condition is (0, 0), (1, 0), (0, 1).

Step 3: DNA tetrahedron probes are placed in the data pool, and the strands satisfying the first constraint are hybridized with the probes to form double strands (as follow Fig. 5). The probe is taken out and cleaned to form a new data pool. The data pool thus formed satisfies the first constraint.

Step 4: repeat step 3, so we will find DNA stands that satisfy all constraints.

Step 5: the extracted DNA stand was purified and amplified by PCR. Finally, the feasible solution is read out. For all feasible solutions, we compare the value of the objective function. Finally, the optimal solution is obtained.

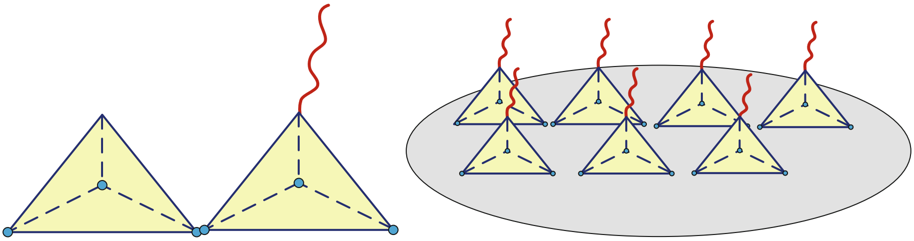


Fig. 4. DNA tetrahedral probe

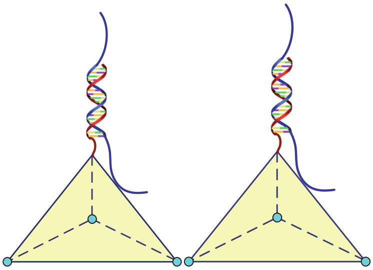


Fig. 5. Extraction of DNA stands satisfying constraints by using DNA tetrahedron probe

Each integer linear programming is equivalent to a 0-1 integer linear programming. The general integer linear programming can be transformed into a simple 0-1 integer programming [28] by mathematical methods. For satisfiability problem, it can also be transformed into a 0-1 integer programming problem.

3.4 Complexity Analysis

The complexity of the model generally includes time complexity and spatial complexity. The time complexity is usually related to its biochemical reaction time. And the spatial complexity is generally related to its computational depth. The short strand number used in this model is related to the representation of variables and the number of variables. The computational depth of the model is related to the constraints of the model, so the complexity of the model is linear with the number of variables n. The above analysis shows that the model greatly reduces the complexity of the integer programming problem, and is a more effective algorithm.

4 Conclusion

In this paper, DNA tetrahedron is constructed by origami, and DNA tetrahedron is used as a probe to solve 0-1 integer programming problem. The complexity of computation is linearly related to the number of variables. The model makes full use of the advantages of origami, such as convenient encoding, fast reaction, low cost and so on.

The tetrahedron has the advantages of stable structure, good toughness and compression resistance, simple fabrication process, high yield, abundant functional modification sites, and good biocompatibility. Therefore, the fabrication of probes with this structure can greatly reduce the generation of pseudo-solutions and improve the efficiency of the solution. With the improvement of experimental environment and the development of molecular biology technology, the biological operation of this DNA tetrahedron computing model will be better realized.

Acknowledgment. This project is supported by National Natural Science Foundation of China (No. 61702008, No. 61672001) and Anhui Natural Science Foundation (No. 1808085MF193).

References

1. Seeman, N.C.: Structural DNA nanotechnology: growing along with nano letters. Nano Lett. **10**(6), 1971–1978 (2010)
2. LaBean, T.H.: Nanotechnology: another dimension for DNA art. Nature **459**(7245), 331–332 (2009)
3. Yang, X., Wenzler, L.A., Qi, J., et al.: Ligation of DNA triangles containing double crossover molecules. J. Am. Chem. Soc. **120**(38), 9779–9786 (1998)
4. Winfree, E., Liu, F., Wenzler, L.A., et al.: Design and self-assembly of two-dimensional DNA crystals. Nature **394**(6693), 539–544 (1998)
5. Rothemund, P.W.K.: Folding DNA to create nanoscale shapes and patterns. Nature **440**(7082), 297–302 (2006)
6. Seeman, N.C.: Nucleic-acid Junctions and Lattices. J. Theor. Biol. **99**(11), 237–247 (1982)
7. Winfree, E., Liu, F.R., Sedman, N.C.: Design and self-assemble of two-dimensional DNA crvstals. Nature **394**, 539–544 (1998)
8. Rothemund, P.W., Papadakis, N., Winfree, E.: Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol. **2**, 2041–2053 (2004)
9. Sa-Ardyen, P., Vologodskii, A.V., Seeman, N.C.: The flexibility of DNA double crossover molecules. Biophys. J. **84**, 3829–38371 (2003)
10. Fu, T.J., Seeman, N.C.: DNA double-crossover molecules. Biochemistry **32**, 3211–3220 (1993)
11. Li, X.J., Yang, X.P., Oi, J., et al.: Antiparallel DNA double crossover molecules as components for nanoconstruction. J. Am. Chem. Soc. **118**, 6131–6140 (1996)
12. LaBean, T.H., Yan, H., Kopatsch, J., et al.: Construction, analysis, ligation, and self-assembly of DNA triple crossover complexes. J. Am. Chem. Soc. **122**, 1848–18601 (2000)
13. Yan, H., Park, S.H., Finkelstein, G., et al.: DNA-templated self-assembly of protein arrays and highly conductive nanowires. Science **301**, 1882–1884 (2003)
14. He, Y., Chen, Y., Liu, H.P., et al.: Self-assembly of hexagonal DNA two-dimensional (2D) arrays. J. Am. Chem. Soc. **127**, 12202–122031 (2005)
15. He, Y., Ye, T., Su, M., et al.: Hierarchical self-assembly of DNA into symmetric supramolecular polyhedral. Nature **452**, 198–201 (2008)
16. Zhang, F., Nangreave, J., Liu, Y., et al.: Reconfigurable DNA origami to generate quasifractal patterns. Nano Lett. **12**, 3290–3295 (2012)
17. Wei, B.R., Dai, M.J., Yin, P.: Complex shapes self-assembled from single-stranded DNA tiles. Nature **485**, 623–626 (2012)

18. Douglas, S.M., Chou, J.J., Shih, W.M.: DNA-nanotube-induced alignment of membrane proteins for NMR structure determination. Proc. Natl. Acad. Sci. USA **104**, 6644–6648 (2007)
19. Andersen, E.S., et al.: DNA origami design of dolphin-shaped structures with flexible tails. ACS Nano **2**(6), 1213–1218 (2008)
20. Dietz, H., Douglas, S.M., Shih, W.M.: Folding DNA into twisted and curved nanoscale shapes. Science **325**(5941), 725–730 (2009)
21. Ke, Y., Sharma, J., Liu, M., Jahn, K., Liu, Y., Yan, H.: Scaffolded DNA origami of a DNA tetrahedron molecular container. Nano Lett. **9**(6), 2445–2447 (2009)
22. Han, D., et al.: DNA gridiron nanostructures based on four-arm junctions. Science **339**(6126), 1412–1415 (2013)
23. Kim, K.R., Kim, D.R., Lee, T., Yhee, J.Y., Kim, B.S., Abn, D.R.: Drug delivery by a self-assembled DNA tetrahedron for overcoming drug resistance in breast cancer cells. Chem. Commun. **49**(20), 2010–2012 (2013)
24. Williams, S., Lund, K., Lin, C., Wonka, P., Lindsay, S., Yan, H.: Tiamat: a three-dimensional editing tool for complex DNA structures. In: Goel, A., Simmel, F.C., Sosík, P. (eds.) DNA 2008. LNCS, vol. 5347, pp. 90–101. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03076-5_8
25. Zhu, J., Wei, B., Yuan, Y., et al.: UNIQUIMER 3D, a software system for structural DNA nanotechnology design, analysis and evaluation. Nucleic Acids Res. **37**(7), 2164 (2009)
26. Goodman, R.P.: NANEV: a program employing evolutionary methods for the design of nucleic acid nanostructures. Biotechniques **38**(4), 548–550 (2005)
27. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bull. Am. Math. Soc. **64**, 275–278 (1958)
28. Wang, S.Y., Yang, A.M.: DNA solution of integer linear programming. Appl. Math. Comput. **170**, 626–632 (2005)
29. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science **266**(11), 1021–1024 (1994)
30. Yin, Z.X., Zhang, F.Y., Xu, J.: The general form of 0-1 programming problem based on DNA computing. Biosystems **70**(1), 73–79 (2003)
31. Zhang, F.Y., Yin, Z.X., Xu, J.: Application of DNA chip on 0-1 planning problem. Biochem. Biophys. **30**(3), 412–415 (2003)
32. Yin, Z.X., Zhang, F.Y., Xu, J.: 0-1 DNA computing model for programming problem. J. Electron. Inf. **15**(1), 1–5 (2003)
33. Wang, L., Lin, Y.P., Li, Z.Y.: DNA computation for a category of special integer programming problem. J. Comput. Res. Dev. **42**(8), 1431–1437 (2005)
34. Zhou K., Tong X.J., Xu J.: The improvement on algorithm of DNA computing on 0-1 programming problem. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, pp. 4282–4286 (2006)
35. Yang, J., Yin, Z.: 0-1 integer programming problem based on RecA- mediated triple-stranded DNA structure. Comput. Eng. Appl. **44**(2), 76–79 (2008)
36. Zhang, X.C., Niu, Y., Cui, G.Z., et al.: Application of DNA self-assembly on 0-1 integer programming problem. J. Comput. Theor. Nanosci. **7**(1), 165–172 (2010)
37. Li, F., Liu, J., Li, Z.: DNA computation model based on self-assembled nanoparticle probes for 0–1 integer programming problem. Math. Comput. Simul. **151**, 1–4 (2017)
38. Yin, Z., Cui, J., Yang, J.: Integer programming problem based on plasmid DNA computing model. Chin. J. Electron. **26**(6), 1284–1288 (2017)
39. Chen, Y.H., Sha, S.: Molecular beacon model of 0-1 integer programming based on microfluidic chip. J. Guangdong Polytech. Norm. Univ. **2**, 004 (2016)