Designing DNA code: quantity and quality

Kaiqiang Liu and Bin Wang*

Key Laboratory of Advanced Design and
Intelligent Computing,
Ministry of Education,
Dalian University,
Dalian, China
Email: liukaiqiang94@163.com
Email: wangbinpaper@gmail.com
*Corresponding author

Abstract: With the development of modern technology, traditional computers have reached the physical limit in information storage and parallel computing. DNA computing was born to solve the above problem. DNA computing is essentially implemented based on a new computational model of DNA molecules. DNA code design is a vital step of DNA computing process. The quantity and quality of the code can directly affect the accuracy and efficiency of DNA computing. At present, the research direction of DNA code is divided into set design and quality optimisation. Set design is to improve the lower bound of the code. Quality optimisation makes DNA code meet more constraints. This paper summarises the code problems in the DNA computing process, the combinatorial constraints that the code needs to satisfy and the intelligent optimisation algorithm used in code. Finally, the problems of DNA code and the future development direction are analysed.

Keywords: DNA computing; DNA code; combinatorial constraints; intelligent optimisation algorithm.

Reference to this paper should be made as follows: Liu, K. and Wang, B. (2019) 'Designing DNA code: quantity and quality', *Int. J. Adaptive and Innovative Systems*, Vol. 2, No. 4, pp.298–323.

Biographical notes: Kaiqiang Liu, majoring in Computer Science and Technology, is currently working toward the his MS degree at Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian 116622, China. His current research interests are DNA sequence design and intelligence computing.

Bin Wang received his BS degree in Computer Science and Technology from Dalian University, in June 2006 and the PhD in Mechanical Design and Theory from Dalian University of Technology in October 2013. He is an Associate Professor at Dalian University. His research areas include intelligence computing, DNA sequence design, DNA cryptography and biological network. So far, he has (co)authored about 41 published papers.

1 Introduction

In Head (1987) proposed the idea of using DNA molecules to compute. Limbachiya et al. (2016) also mentioned in his literature that Tom Head gave the idea of using DNA computing. Professor Adleman of the University of Southern California used DNA molecules as the computational medium to successfully solve the directional Hamilton path problem of seven vertices in the biological laboratory by using modern biotechnology in 1994. For more details, reader can refer to Adleman (1994) and Garzon (1997). The advent of Professor Adleman's research has immediately attracted worldwide attention from scientists in the fields of mathematics, physics, chemistry, biology and computer science. In 1995, Professor Lipton of Princeton University published an article in 'Science' inspired by Professor Adleman's thought to further demonstrate the complete problem of DNA computing (Lipton, 1995).

Then more than 200 experts from around the world discussed the feasibility of DNA computing and optimistically believe that once the DNA computer is successfully developed, its computing power is beyond the reach of traditional computers, which is a field with great science, engineering, social and economical value. Compared with traditional computers, DNA computers have the following outstanding advantages:

1 Highly parallel: the amount of computation in a DNA computer is the total amount of computation since the advent of traditional computers.

2 Mass storage capacity: 1 cubic metre DNA solution can store 1,023 binary numbers.

3 Low energy consumption: the energy consumed by DNA computer is 10–10 of the energy consumed by a traditional computer to perform the same computing.

4 Abundant resources: the technology for extracting DNA from plants and animals is very mature and animals and plants in nature can be seen everywhere.

DNA is composed of three components: nitrogen base, sugar and phosphoric acid. The nitrogen-containing base components in DNA are adenine A, guanine G, cytosine C and thymine T. The DNA sequence is made with series of these four letters, AGCT. It is an ideal material for DNA computing due to its density, stability and self-replication properties. The basic idea of DNA computing is to use the special double helix structure of DNA molecules and the principle of base pair matching to encode information. In general, DNA Computing includes the following three steps:

1 Encode: maps the original problem to be solved into a collection of DNA molecules.

2 Computing process: all biochemical reactions are carried out to generate all possible solution spaces with the participation of related enzymes.

3 Separation and reading of the solution.

Designing reliable DNA sequences is an important task in DNA computing. Hybridisation between a sequence and a complement sequence is the most core reaction in DNA Computing. The efficiency and precision of the hybridisation reaction directly affect the efficiency of DNA Computing and the reliability of the final output. At present, there are two main aspects in DNA Computing to ensure the smooth progress of the Computing: one is to optimise each information element in DNA computing by code; the other is to improve the reliability and accuracy of biochemical operation. The purpose of the code problem in DNA computing is to hope that in the DNA computing process, the DNA molecules encode each information element can be uniquely identified, that is, completely hybridised, thereby minimising the occurrence of non-specific DNA hybridisation (Tulpan, 2006).

At present, DNA computing research has been applied in many aspects such as: DNA tile components (Winfree, 1998; Winfree et al., 1998), the establishment of DNA nanostructures (Rothemund, 2006; Seeman, 1998), to study the error correction characteristics in DNA sequences (Arita, 2004; D'Yachkov et al., 2003), DNA-based data storage systems (Limbachiya and Gupta, 2015; Garzon et al., 2006), molecular barcode (Tulpan et al., 2002; Wang et al., 2018b), biomolecular Computing (Frutos et al., 1997), DNA microarrays (Graves, 1999; Pirrung, 2002), image encryption (Wang et al., 2018a), PCR amplification (Dan et al., 2012; Rose et al., 2003).

RNA computing is a molecular computing model similar to DNA computing (Chidchob and Sleiman, 2018). Faulhammer et al. (2000) extended the idea of DNA computing to RNA and proposed a general RNA computing way to solve SAT problems. Due to the new discovery of RNA thermodynamic stability (Dineen, 2011), RNA nanoparticles are becoming another promising computing medium for nanotechnology and nanomedicine (Qiu et al., 2013). Recently, nanotechnology plays an important role in medicine (Ma et al., 2018) and gene logic chip (Masubuchi et al., 2018). RNA molecular is not easy to operate in vitro and is not convenient to process information like DNA molecular (Jin, 2014). More details about RNA computing can refer to Cukras et al. (1999).

This paper provides a detailed summary of DNA code algorithms in terms of both quantity and quality optimisation. The main research direction of designing set problem in recent years is to improve the lower bound of code. The most important contribution to optimisation is the multi-objective optimisation problem. Among the algorithms for designing DNA code listed in this paper, the exhaustive method is the most direct and least complex algorithm, but this type of algorithm is not very efficient because it consumes a lot of computing resources. Evolutionary algorithms and improved evolutionary algorithms have been widely used in the design of sequences.

The rest of the paper is organised as follows. In Section 2, details the DNA code problem. In Section 3, we introduce the algorithms for designing DNA code. Finally, the conclusions and future works are included in Section 4.

2 DNA code problem

The code problem is the first and most important step in DNA computing. The so-called code problem is to design an isometric DNA code that satisfies certain combinatorial constraints with four bases A, T, C and G. The quality of the code directly affects the speed and efficiency of the reaction process. The purpose of the code problem research in DNA computing is to reduce the probability of false hybridisation as much as possible in the biochemical reaction process and to improve the efficiency and reliability of DNA computing.

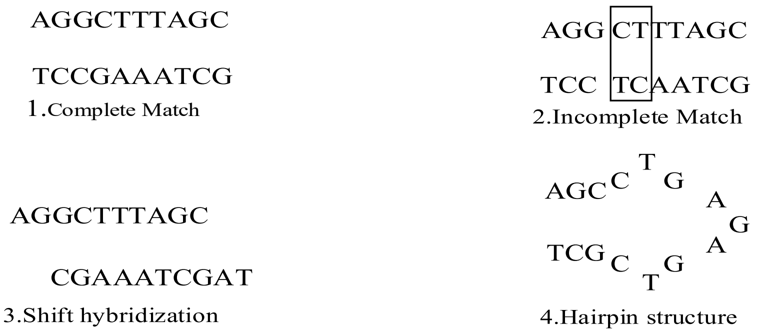
In the process of hybridisation of DNA molecules, there are generally two types of non-specific hybridisation forms (Deaton et al., 1997; Deaton and Garzon, 1998):

1 False positive: DNA molecules that are not completely complementary hybridise under appropriate conditions to form double-stranded molecules, which is mainly. The two DNA molecular sequences of hybridisation are caused by sufficient ‘similarity’.

2 False negative: completely complementary DNA molecules do not hybridise as expected during the reaction, mainly due to reaction conditions and biochemical operations.

Caused by its own mistakes. Figure 1 shows several hybridisation methods that may exist between DNA molecules.

Figure 1 Several possible hybridisation methods of DNA molecular sequences



Therefore, reasonable code can minimise the occurrence of false positives and also contribute to the occurrence of false negatives.

2.1 Definition of code problem

The definition of code problems in DNA computing w and as first mentioned by Garzon (1997). After refining, they gave a complete definition of the code problem in DNA Computing in the literature Garzon and Deaton (2004). An example of an algorithmic problem is systematically mapped to a particular sequence of DNA molecules that are capable of ensuring that subsequent biochemical reactions are carried out without any errors. Moreover, the product after biochemical reaction must contain enough, stable and reliable solutions to the original problem that can be successfully extracted in Garzon and Deaton (2004). In formal language, it can be expressed as: on the four bases of the DNA molecule $\sum\{A, G, C, T\}$, there is a code set S of DNA molecules of length m. Find a subset of S called C, So that for $\forall S_i, S_j \in C$ is satisfied.

$$\tau(S_i, S_j) \geq K \tag{1}$$

Here, *K* is a positive integer and is an evaluation criterion of the code property, such as a Hamming distance, GC content, a shift distance and a minimum number of identical subsequences.

Followed studies have shown that in the study of DNA code problems, two indicators are mainly concerned: code quantity and quality. The higher the code quality, the higher the reliability of the computing; the larger the number of code, the larger the problem-solving scale. However, in practical problems, these two indicators are contradictory: the higher the code quality, the less the number of code. Therefore, this paper summarises the work related to DNA code in recent years from the perspective of code quality and number of code. That is, under the condition that a certain code quality is satisfied, the obtained maximum code sequence set is obtained; when a certain number of codes is satisfied, the quality of the code is optimised.

2.2 DNA code constraints

There are many constraints that DNA code needs to satisfy, which can be divided into the following three categories:

1 Eliminate similarity constraints.

2 Thermodynamic constraints.

3 Application-oriented constraints.

2.2.1 Eliminate similarity constraints

2.2.1.1 Hamming distance

Given two encode of the same length $X = X_1, X_2, \dots X_n$ and $Y = Y_1, Y_2, \dots, Y_n$. Here, $H(X, Y) \geq d$ is used to indicate that the Hamming distance constraints is satisfied between the code X and the code Y and the two code are different (Kawashimo et al., 2006a).

$H(X, Y)$ represents the total number of coded and encoded coded characters on all corresponding coded bits, that is (Chee and Ling, 2008):

$$H(X,Y) = |\{1 \leq i \leq n : X \neq Y\}| \tag{2}$$

2.2.1.2 Reverse hamming distance

Given two encode of the same length $X = X_1, X_2, \dots, X_n$ and $Y = Y_1, Y_2, \dots, Y_n$. $(X)^r = X_n, \dots, X_2, X_1$ denotes the inverse sequence of the code. Here, using $H(X^r, Y) \geq d$ denotes code X and Y satisfied inverse Hamming distance and the two code are the same. That is:

$$H\left(X^r, Y\right) = |\{1 \leq in : X^r \neq Y\}| \tag{3}$$

2.2.1.3 Complementary hamming distance

Given two encode of the same length $X = X_1, X_2, \dots, X_n$ and $Y = Y_1, Y_2, \dots, Y_n$. Using $X^c = X_1^c X_2^c, \dots, X_n^c$ represents the Complementary Hamming distance of X. Using $H((X)^c, Y) \geq$ d represent code X and code Y satisfied complementary hamming distance constraints, this two code can be same. That is (Tanaka et al., 2002):

$$H\left((X)^c, Y\right) = \left|\{1 \leq i \leq n : (X)^c \neq Y\}\right| \tag{4}$$

2.2.1.4 Reverse complement hamming distance

Given two encode of the same length $X = X_1, X_2, \ldots, X_n$ and $Y = Y_1, Y_2, \ldots, Y_n$. Code $(X)^{rc}$ represents Reverse Complement Hamming distance of X. Using $H((X)^{rc}, Y) \geq d$ represents X and Y satisfied Reverse Complement Hamming distance, this two code are the same. That is (Chee and Ling, 2008):

$$H\left((X)^{rc}, Y\right) = \left|\left\{1 \leq i \leq n : (X)^{rc} \neq Y\right\}\right| \tag{5}$$

The researchers (Cervantes-Salido et al., 2013; Santalucia, 1998) considered that a suitable inverse complement Hamming distance constraints would reduce the non-specific hybridisation between the inverse sequence of the code sequence and the complement of the code sequence to some extent.

2.2.1.5 H-measure

Garzon propose an H-distance criterion in Xiao et al. (2013), which is defined as the minimum of the Hamming distances obtained by moving any two sequences by $k(-n < k < n)$ distances. In theory, Garzon's H-distance criterion can minimise the similarity between two codes, but its computational complexity greatly increases the computational difficulty of code problems (Banzhaf et al., 2000).

2.2.1.6 Hairpin structure

If the DNA code sequence designed in the biochemical reaction process is partially hybridised, a hairpin structure will be formed (Shin et al., 2005). The mathematical expression of the hairpin structure is as follows:

$$f_{Hairpin}(\Sigma) = \sum\nolimits_{i=1}^{n} Hairpin(\Sigma_i) \tag{6}$$

$$Hairpin(\Sigma) = \sum\nolimits_{p=P_{\min}}^{l-R_{\min}} \sum\nolimits_{r=R_{\min}}^{1-2p} T\left(\sum\nolimits_{j=0}^{pinlen(p,r,i)} bp\left(x_{p+i-j}, x_{p+i+r+j+1}\right), \frac{pinlen}{2}\right) \tag{7}$$

The hairpin structure constraints can effectively avoid the formation of the secondary structure of the same DNA code sequence in the hybridisation reaction to a certain extent and allows the specific hybridisation to be carried out efficiently, thereby ensuring the smooth operation of the DNA computing.

2.2.1.7 Continuity

If the designed DNA code often contains many consecutive identical bases, then the internal code of the DNA will be hydrogen bonded to form a secondary structure in the hybridisation reaction (Shin et al., 2005). The DNA code continuity (Cervantes-Salido et al., 2013) is constrained as follows:

$$Con(\alpha) = \sum\nolimits_{i=1}^{m} \sum\nolimits_{j=1}^{n} (j-1) N_j^{(i)} \tag{8}$$

Here $N_j^{(i)}$ represents the total number of times the same coded character in the DNA code α is repeated j times.

2.2.2 Thermodynamic constraints

2.2.2.1 Melting temperature

The melting temperature T_m is the temperature at which 50% of the DNA double-stranded molecules become single-stranded during DNA denaturation. It is an important parameter to determine the efficiency of the reaction. Generally, the melting temperature T_m of the DNA molecules participating in the reaction is substantially the same. The factors affecting the melting temperature T_m are: the composition, concentration and pH of the DNA molecule. The T_m formula proposed by Santalucia (2013) based on the Breslauer scheme is:

$$T_m = \frac{\Delta H}{\Delta S + R \times \ln(c/4)} - 273.15 \tag{9}$$

where R is the molar gas constant, C is the nucleic acid concentration and ΔS and ΔH represent the entropy change and enthalpy change between the bases at a certain temperature, respectively. After ΔS and ΔH are determined, the formula $\Delta G = \Delta H - T \times \Delta S$ derives the free energy change between the bases.

2.2.2.2 Free energy

Free energy change is an important parameter for evaluating the thermodynamic stability of DNA molecules. The hybridisation reaction between any two DNA molecules can be expressed by the following chemical equation:

$$X + Y \leftrightarrow XY \tag{10}$$

where in XY represents the double strand after hybridisation. It is known from chemical thermodynamics that the direction of the hybridisation reaction is the direction in which the free energy is reduced. Free energy is the energy released by a single-stranded DNA molecule participating in a chemical reaction from a high-energy state to a low-energy state double-stranded molecule. The free energy is calculated using the nearest-neighbour thermodynamic model in Kawashimo et al. (2006b). The formula is as follows:

$$\Delta G = \sum_i n_i \Delta G(i) + \Delta G(iniGC) + \Delta G(iniAT) + \Delta G(sym) \tag{11}$$

In the above formula, $\Delta G(i)$ represents the free energy of adjacent base pairs, for example: $\Delta G(1) = \Delta G(AA / TT)$, $\Delta G(2) = \Delta G(TA / AT)$, a total of ten species of adjacent base pairs of Waston-Crick, where i is the number of $\Delta G(i)$, $\Delta G(iniGC)$ is the correction value of GC pairing at the starting position and $\Delta G(iniAT)$ is the correction of the AT position of the starting position. The value, $\Delta G(sym)$, is the corrected value of the self-complementing DNA sequence , reader can refer to Tulpan et al. (2015).

2.2.2.3 Constant GC content

In DNA, the bases C and G in the deoxyribonucleic acid pair, three hydrogen bonds are formed between them and two hydrogen bonds between the paired bases A and T generate two hydrogen bonds and connect them together. The number of hydrogen bonds in DNA directly reflects its thermal stability. The higher the GC content, the higher the number of hydrogen bonds and the higher the thermal stability of DNA molecules (Chee and Ling, 2008), so good GC content constraints are designed. An important guarantee for good thermal stability DNA code. In general, the *GC* content constraints means that for any one code α, the total number of characters *G* or *C* in the code α is half the code length [*n*/2]. In this paper, *GC*(α) is used to indicate the *GC* content of α, which is (Chee and Ling, 2008):

$$GC(\alpha) = \left|\{1 \le i \le n : \alpha_i \in \{G, C\}\}\right| \tag{12}$$

In the formula (9), the subscript *i* indicates any one of the code α.

2.2.3 Application-oriented constraints

Irrelevant constraints: Any code sequence X in a DNA sequence set moves to the right space (m ≤ n) and does not hybridise with other remaining sequences. The DNA sequence X and sequence Y irrelevant constraints are denoted as X°Y, for example X = CATCATC, Y = ATCATCGG, X°Y = 0100100. Figure 2 is The Computing process.

Figure 2 The computing process

```
X   =   C  A  T  C  A  T  C
Y   =   A  T  C  A  T  C  G  G                       0
Y   =      A  T  C  A  T  C  G  G                    1
Y   =         A  T  C  A  T  C  G  G                 0
Y   =            A  T  C  A  T  C  G  G              0
Y   =               A  T  C  A  T  C  G  G           1
Y   =                  A  T  C  A  T  C  G  G        0
Y   =                     A  T  C  A  T  C  G  G     0
```

3 Algorithms for the DNA code

3.1 Algorithms for designing DNA code set

Under the condition that a certain number of code is satisfied, the largest set of code sequences is required, which may be called a DNA code set design. The main research work of DNA code set design is to improve the lower bound of code (Chee and Ling, 2008). There are many algorithms for designing DNA code set. Such as the algorithm of code theory, but this algorithm is hindered by the complexity of combinatorial constraints, it is often difficult to use theoretical reasoning. Therefore, it is mainly a heuristic intelligent optimisation algorithm. This paper systematically summarises the algorithms used for designing DNA code set.

3.1.1 Genetic algorithm

Deaton first applied genetic algorithm in 1996 to design a DNA sequence that satisfies the Hamming distance and the Reverse complement Hamming distance and found its upper bound value, (Deaton, 1996) can be referred. They used an accurate experiment to analyse the performance of the sequence generated by this algorithm for DNA Computing. In Zhang (1998) used an iterative genetic search to design DNA sequences in

1998. Deaton used genetic algorithms to design a code set that avoids false positives in the reaction to solve the Hamiltonian problem in 1999. Arita et al. (2000) designed a DNA sequence using a genetic algorithm that satisfies three fitness functions in 2000, can be referred. They designed a self-complementary code for the Whiplash model and compared the results with a randomly generated test algorithm. The results of repeated experimental simulations show that the algorithm obtains the lower bound value better than the experimental results provided by the previous literature.

3.1.2 Stochastic local search algorithm

Stochastic local search algorithm was first described in Tulpan et al. 2002 was used to design a set of DNA code. In 2003 used the hybridisation randomisation neighbourhood to improve the stochastic local search algorithm in the literature (Tulpan and Hoos, 2003). In Tulpan (2006) used an effective heuristic algorithm to design DNA sequences in his PhD thesis. The Stochastic local search algorithm is suitable for the search space of infeasible solutions. This algorithm starts with a given large number of DNA sequences that satisfy the combined constraints. There is fitness metric to estimate the severity of the solution's violation of the overall constraints. The algorithm iteratively modifies DNA sequences with the goal of improving fitness. Finally, if all the constraints are met, then a feasible solution is found; otherwise, the smallest set of DNA sequences that caused the violation can be deleted, so that there is a feasible solution. In 2008, Chee and Ling introduced a group search in stochastic local search to make the lower bound value of the generated DNA set greatly improved.

3.1.3 Linear code-based algorithm

Gaborit et al. first proposed using a linear code algorithm in 2004 to construct DNA code, (Gaborit and King, 2004) can be read. In his paper, a new construction algorithm for DNA code is proposed, which requires that the code satisfies the reverse complement constraints, the GC content constraints, or both. Aboluion et al. (2011) made further improvements based on this idea in 2010.

1 Linear code based on GF(4): in Varbanov et al. (2015) this algorithm maps the elements in GF(4) to nucleotides in the DNA molecule sequence using different one-to-one mappings to construct a DNA code. The preferred mapping is from $\{0, 1, \omega, \omega^2\}$ to {A, T, C, G}, reader can refer to Dan et al. (2014) and Macwilliams and Sloane (1997). This algorithm yields some of the best lower bounds for GC-constrained code sets and further enriches the design of DNA-encoded sets.

2 Extension based on GF(4), additional and additional extension code: With the different mapping from domain or ring to DNA code, based on extension, addition and additional extension code and using computer algebra systems to improve the lower bound of the code set in Aboluion (2012).

3 Ring-based code: algebraic DNA code construction further extends ring-based code in Rocha et al. (2010). In recent years, more research has been done to map high-quality DNA code sets based on different loops by mapping loops to DNA code.

4 Algebraic number theory code: the algorithm (Hong et al., 2016) uses algebraic number theory to obtain a coded set with a larger code length by means of irreducible cyclic code.

3.1.4 Template mapping

The algorithm of template mapping was originally proposed by Frutos et al. (1998) from Wisconsin University in 1998. They are mainly engaged in research on DNA computing based on surface algorithm. The template mapping algorithm is to divide the encode process of DNA molecules into two steps:

1 Search for a binary string satisfying certain conditions as a template set T, where '1' represents the position of A/T and '0' represents the position of G/C.

2 Searching for a binary string satisfying certain conditions as a mapping set M and finally obtaining a desired set of DNA encode sequences.

Japan's Arita and Kobayashi (2002) further studied the template mapping algorithm in 2000. In order to overcome the shift hybridisation between DNA sequences, they introduced a minimum distance definition. Arita et al. (2002) proposed a template mapping strategy in 2002 to select a large number of dissimilar sequences by using only a small number of templates and mappings. They built some pre-designed templates and then statistically generated a set of sequences to ensure a small probability of mismatch. The advantage of the template approach is that it can find a reliable sequence in a short time within a given error rate. In the same year, Arita et al. (2002) used template mapping to generate DNA sequences.

In this paper, a design algorithm using two binary words, namely a template and an error correction codeword, is introduced. The basic idea is to design the sequence in two steps by fixing the [AT] and [GC] positions of all sequences. In the first step, the positions of [AT] and [GC] are determined. These locations are specified by template x_1, x_2, …, x_i. Where 1 represents [AT] and 0 represents [GC] (and vice versa). In the second step, select A or T as the template position $x_i = 1$ and select G or C as the position. These choices are specified by the error correction codeword. Liu et al. (2003) also used template mapping in 2003 to design DNA code. In this document the authors combine template strategies with H-measure. Experimental simulation results show that H-measure can be used to optimise templates and mapping sets more easily and more importantly, it improves the quality of DNA sequences generated by template strategies.

3.1.5 The Salmon algorithm

The Salmon algorithm was inspired by the behaviour of squid to spawn upstream. But it is not the same as the real world of salmon spawning behaviour, but an idealised imitation, but there are many commonalities (Bodo, 1997).

Ashlock used the Salmon algorithm to design DNA error correction code in 2009, reader can refer to Ashlock and Houghten (2009). This paper addresses the problem of cross-operator utility for optimising DNA error correction code in earlier studies. Most cross-evident events produce violations of the minimum distance constraints required for error correction. A new algorithm, modified evolutionary strategy form, has been tested and found to have a record size record and updated the set size of the DNA error correction code. In Ashlock et al. (2002) introduced a new algorithm to synthesise DNA error correction code. This new algorithm is used by the Salmon algorithm to search for DNA error correction code (Orth and Houghten, 2011). The algorithm is a heuristic intelligent algorithm obtained by simulating Salmon's spawning behaviour, which is similar to the artificial ant colony optimisation algorithm, about ant optimisation can be read in Dorigo et al. (2007). In Orth and Houghten (2011) realised the use of the Salmon algorithm to generate DNA error correction code through experiments and adjusted the specific values of the algorithm parameters through a large number of experiments.

3.1.6 Neighbourhood search algorithm

3.1.6.1 Variable neighbourhood search algorithm

Montemanni and Smith (2008) used a neighbourhood search in 2008 to construct DNA code that satisfies GC content. This algorithm combines four local search algorithms based on the variable neighbourhood search framework. This algorithm was first proposed by Montemanni and Smith (2009b). The first local search algorithm used is the seed construction algorithm (Brouwer et al., 1990). These initialised seed sequences all satisfy a constant GC content and are added to the seed if the necessary combinatorial constraints are met. The second local search algorithm is a group search (Bomze et al., 1999), which mainly deletes a subset of the code set, leaving a partial code set. The third way of local search is hybrid search. The main idea of this algorithm is to combine the ideas of the first two local searches. The last local search algorithm is iterative greedy search (Tulpan et al., 2002), which is different from the previously described local search algorithm. Iterative greedy search is devoted to some inflexible code sets, that is, not all code are compatible with each other according to constraints. This improved version of the variable neighbourhood search has been widely used and the lower bounds of DNA code have been well improved in this experiment. Tulpan in literature also used variable neighbourhood search techniques in 2014 and assembled four different local searches into this framework (Hansen and Mladenović, 2007). Their goal is to change the structure of neighbourhood searches over time (Montemanni and Smith, 2008). Through repeated experimental simulations, this algorithm effectively increases the lower bound value of DNA code.

The variable neighbourhood search algorithm in Montemanni and Smith (2008) uses the advantages of different local search algorithms to search for DNA code sets. Some of the search algorithms currently applied in the variable neighbourhood is as follows:

1 Build a seed search: Montemanni and Smith (2008) first introduced four local search algorithms in 2008 and combined the variable neighbourhood search framework. Followed, in Montemanni and Smith (2009a) designed a binary code that satisfies constant weight according to this idea. In the same year, Montemanni et al. (2009) designed the DNA sequence to meet specific GC content in the literature. This algorithm is suitable for the search space of a viable solution. The algorithm uses a set of null DNA code and as a set of repeated iterations to increase the encode of DNA sequences. The set of DNA encode that are added to the collection are collections of code that satisfy the constraints.

2 Clique search: the problem of code can be addressed by solving the problem of the largest group in Bomze et al. (1999). This search algorithm is to create a graph of

compatible sets of code. The vertices are represented by code and the pair of vertices is connected if and only if the two vertices in the figure satisfy a particular combinatorial constraints. After the graph is established, the heuristic algorithm or the determined algorithm is used to combine the coded set obtained by the algorithm with the previous code set. If the code set obtained after iteration is completed is larger than the current best code set, the set is retained.

3 Hybrid search: the hybrid search uses the coded set constructed by the seed as input to the group search and then continues to search for the optimal code using the group search.

4 Greedy approach: according to reference Tulpan et al. (2002) a greedy search will discard the poorer code from the set every iteration.

3.1.6.2 Dynamic neighbourhood search algorithm

A dynamic neighbourhood search algorithm was proposed by Kawashimo to design DNA code that satisfy thermodynamic combinatorial constraints in 2007, reader can refer to Kawashimo et al. (2006b). The research work of this paper considers the DNA sequence design algorithm under thermodynamic constraints from the perspective of combinatorial optimisation. More specifically, a local search type algorithm is used for thermodynamic DNA sequence design. Local search is a way to find a good solution by replacing the current solution with a better solution until a better solution is not found (Kawashimo et al., 2006a. The algorithm can dynamically change the neighbourhood structure and has high search performance. Experiments show that the algorithm has good design ability and successfully produces a better sequence set than the previous algorithm Arita et al. (2002). The simulation results show that the algorithm has good flexibility, can adjust the constraints well and design thermodynamic based DNA code.

3.1.7 Improved leapfrog algorithm

The hybrid frog leaping algorithm (SFLA) is a group-based sub-heuristic collaborative search group intelligent algorithm proposed by Eusuff and Lansey (2003). The hybrid frog leaping algorithm is based on the meme evolution of individuals in the population and the use of memes to achieve global information exchange. The Hybrid Frog Leaping Algorithm (SFLA) combines the advantages of genetic algorithm-based meme calculus algorithm (MA) and particle swarm optimisation algorithm based on group foraging behaviour (PSO). It has simple concept, few parameters, fast computing speed and global the ability to find superior ability, easy to achieve and so on.

The frog leaping algorithm generates population P of N individuals in the D-dimensional space. According to the evaluation rules, the individuals in the population are arranged in descending order of fitness. According to the evolutionary thought of the memes, the frog populations sorted in descending order are divided into X meme groups. Y individuals per memetic group and. The worst individuals in each memetic group are updated, so that information is exchanged within each memetic group and a global search for the optimised solution is completed in the local search process.

A leapfrog algorithm is used to construct a DNA code set that satisfies the Combinatorial of Hamming distance, inverse complement Hamming distance and fixed GC content. The improved frog leaping algorithm is used to initialise and generate

20 DNA code as the initial population. The code of the initial population itself satisfies the fixed GC content and the inverse complement Hamming distance. Next, the populations are grouped according to the memetic group and the information exchange is performed in the meme group to complete the global search of the optimal solution. With this improved leapfrog algorithm, the lower bounds of the previous code set are well improved.

3.1.8 Quantum evolutionary algorithm

Narayanan and Moore (1996) first introduced the idea of quantum into genetic algorithm in 1996 and proposed a quantum heuristic genetic algorithm. Compared with traditional genetic algorithms, quantum genetic algorithms can maximise the diversity of populations. In Guo et al. (2017) proposed a quantum chaos algorithm based on Bloch spherical coordinates to design DNA code set. The algorithm uses the insect model in the chaotic equation to initialise the qubits in the Bloch sphere coordinates. Then the full interference crossover strategy is used for cross operation, then the quantum non-gate is used for mutation operation and finally the dynamic rotation strategy is used to adjust the quantum rotation angle.

Based on the Block quantum chaos algorithm, a chaotic system, that is, an insect model, also called a logistics map, is initialised to generate r random variables. The qubits of the population Bloch spherical coordinates are initialised using the generated r random variables. The algorithm uses a quantum revolving door to update the population, so that each chromosome in the population gradually approaches the optimal chromosome in the population under the action of the quantum revolving door. This process will generate a large number of new chromosomes and the population will evolve. In order to solve the problem that the algorithm falls into local optimum, the algorithm introduces the intersection of quantum chromosomes, which improves the diversity of the population and maintains the better individuals in the group. The intersection algorithm adopted in this paper is not the intersection of traditional genetic algorithms, but the full interference intersection of all chromosomes in the population participating in information exchange. Such crossover greatly increases the diversity of populations. The variation of quantum chromosomes is done by means of NAND gates.

3.1.9 Other algorithm

The easiest way to design DNA sequences is to exhaustive search and random search. Hartemink et al. used an exhaustive algorithm to generate DNA sequences in 1998, refer to Hartemink et al. (1999) and Hybridisation et al. (1997). According to Penchovsky and Ackermann (2003) designed DNA sequences in 2003 using a random search algorithm.

The simulated annealing algorithm is a meta heuristic algorithm derived from the thermodynamic principle. It was first proposed by Kirkpatrick in 1983 and applied to combinatorial optimisation problems (Kirkpatrick et al., 1983). Tanaka et al. (2001) provided some sequence adaptation criteria and generated sequences using simulated annealing. They also tried to find the appropriate Combinatorial of fitness functions to find a viable solution.

The ant colony algorithm is a population-based heuristic bionic evolution algorithm proposed by the Italian scholar Colorni (1992) and Dorigo (1992) in the early 1990s by simulating the behaviour of ants collective path finding in nature. The hybrid search algorithm first initialises a certain number of DNA code sets V (the code set satisfies a certain GC content). The ant k is used to construct a DNA code set S_k that satisfies the constraints condition. The ant k randomly selects a code from the code set V, adds it to the set S_k and then continuously expands the set S_k and selects the code rule to obey the roulette principle. The process is iterated until the candidate solution is an empty set and the ant k constructs the DNA code set S_k.

3.2 Algorithms for optimising DNA code

The expansion of the number of code is conducive to solving the problem of large scale, but the optimisation of the code quality can improve the accuracy of the hybridisation reaction. The optimisation problem of code quality is generally divided into single-objective optimisation and multi-objective optimisation. The so-called single-objective optimisation is to convert the multi-objective problem into a single-objective problem by adding different combinatorial constraints to generate a fitness function through the generated weights. This paper provides a detailed review of the quality optimisation algorithms used for DNA code.

3.2.1 Genetic algorithm

Genetic algorithms combined with many emerging intelligent optimisation algorithms produce many improved hybrid intelligent optimisation algorithms. These improved algorithms have been very good for quality optimisation of DNA code and optimisation of multi-objective problems.

Quantum genetic algorithm is a new optimisation algorithm that combines quantum computing and genetic algorithm. In recent years, many scholars have used particle swarm optimisation to automatically follow the new quantum angular size. At the same time, in order to overcome the shortcomings of the particle swarm optimisation algorithm, it is easy to fall into the local optimal solution (Xiao et al., 2009). The commonly used Tent mapping chaotic search is introduced into the quantum genetic algorithm and a new hybrid quantum genetic algorithm is proposed and applied to DNA code. Sequence optimisation problem. By incorporating particle swarm optimisation and Tent mapping, the algorithm can not only have fast convergence, but also overcome the shortcomings of the algorithm being easy to fall into the local optimal solution. Simulation of DNA sequence code problems revealed that efficient DNA sequences were available.

In the same year, Xia proposed a multi-target carrier chaotic evolution to optimise the multi-objective problem of DNA code (Xiao et al., 2009). The traditional weighting algorithm has the disadvantage of difficulty in selecting weights. In the paper, the chaotic search of power function carrier is integrated into the intensity Pareto evolutionary algorithm and the multi-target carrier chaotic evolution algorithm is proposed. It is applied to solve the optimisation problem of DNA sequence and obtain the DNA code sequence with good quality.

Wang proposed a micro-genetic algorithm in reference Peng et al. (2016) to introduce a shared function based on similarity and Hamming distance in 2016. Six design criteria were applied and four genetic operators were applied to obtain higher quality DNA sequences.

NSGA-II (rapid non-dominated genetic sorting genetic algorithm) is a multi-objective optimisation reference algorithm developed by Deb (2000). Chaves-González et al. (2014a) used a fast non-dominated genetic algorithm in 2014 to generate DNA sequences. This sequence generation problem is a multi-objective optimisation problem. The optimisation function requires similarity, H-measure, continuity and hairpin structure to be minimal, while meeting the melting temperature and GC content. The algorithm randomly selects two sequences from the population according to the operation of the genetic algorithm and then merges the parent population with the parent population and then performs fast non-dominated sorting. The resulting sequence is a reliable sequence capable of biocomputing.

3.2.2 Particle swarm optimisation algorithm

Cui et al. (2007) used particle swarm optimisation algorithms in DNA code optimisation problems in his paper. In this paper, a discrete problem continuous strategy is proposed, so that the standard particle swarm optimisation algorithm which can only solve the continuous optimisation problem can be used to solve the DNA quality optimisation problem belonging to discrete problems. A DNA code sequence design is also proposed. A quaternary discrete particle swarm optimisation algorithm. The simulation results show that these two algorithms have a good effect on the code problem of smaller scale.

Xu et al. (2008) treated DNA sequence design problems as multi-objective problems, using genetic algorithms and particle swarm optimisation algorithms to optimise DNA sequences. In this paper, the GA/PSO algorithm is used, which combines the convergence of the genetic algorithm with the fast convergence of the particle swarm optimisation algorithm. This algorithm can enhance the diversity of the population and improve the ability of the seed to converge. In order to avoid the traditional fitness algorithm to ignore the best fitness, the best preservation strategy is adopted here. Better code quality is obtained with the improved algorithm. In the same year, Khalid et al. (2009) and Xu and Zhang (2008) used a binary particle swarm optimisation algorithm to optimise DNA sequences in 2009. This algorithm uses a weighted approach to convert Hamming distance, similarity, continuity and hairpin into a single target problem. Binary particle swarm optimisation is to satisfy the melting temperature and GC content to solve the target problem. After repeated simulation experiments, this algorithm can obtain better quality DNA code sequences than other algorithms.

3.2.3 Invasive weed algorithm

Under the guidance of his tutor Wang, Yang introduced the niche exclusion mechanism into the DNA code sequence optimisation design of the invasive weed algorithm in 2016. Inspired by natural phenomena, according to Mehrabian and Lucas (2006) proposed a new intelligent optimisation algorithm invasive weed optimisation algorithm in 2006. The algorithm is an intelligent optimisation algorithm for simulating the weed colony diffusion process in nature. The invasive weed algorithm gives full play to the 'leadership' role of the outstanding individuals in the population, which guides the evolution of the group. The descendants of the descendants choose to distribute around the parent individuals in a normal distribution. The invasive weed algorithm has strong robustness and adaptability and can converge to the optimal solution of the problem simply and effectively. Jong (1975) first proposed a niche strategy based on crowded wit.

The main idea of the algorithm is to set a crowding factor CF, randomly select 1/CF individuals in the population as the crowding members and replace the newly generated individuals with the crowded members to eliminate some similar individuals according to the similarity. In this paper, the standard invasive weed algorithm is improved. The improved children are distributed around the parent in the Cauchy distribution. And introduce the idea of niche exclusion mechanism. The improved algorithm uses a weight-based fitness function to calculate the fitness value. The DNA code quality generated by the algorithm is higher by analysing the experimental results.

3.2.4 Harmony search

The harmony search algorithm is a novel intelligent optimisation algorithm proposed by Korean scholar Zong et al. (2001). The algorithm simulates the process of music musicians in their music, by repeatedly adjusting the pitch of each instrument in the band and finally achieving a wonderful harmony state. A great feature of the harmony search algorithm is that the initial solution can be given randomly and other heuristic algorithms can be used to construct an initial solution. For complex optimisation problems, a heuristic algorithm should be used to construct a feasible solution as the initial solution.

Kennedy and Eberhart (1995) proposed a particle swarm optimisation algorithm based on the simulation of bird behaviour and fish behaviour in 1995. Particle swarm optimisation is initially dealing with continuous optimisation problems and its applications have been extended to combine optimisation problems. Due to its simple and effective characteristics, the particle swarm optimisation algorithm has been valued and studied by many scholars.

The fitness function of the algorithm is different from the fitness function of the harmony search and particle swarm optimisation algorithm, but a fitness function based on small population is proposed. Thereby greatly improving the optimisation results of the DNA sequence. In each generation of the particle swarm algorithm, the individual performs the harmony search in the late stage, so that each individual searches for the best position around himself and adjusts the population structure. In order to solve the defect that the particle swarm algorithm is easy to fall into the local optimal solution, the algorithm also adds the optimal preservation strategy to ensure the obtained result is the global optimal solution.

3.2.5 Artificial bee colony algorithm

Chaves-González et al. (2013) proposed a multi-target swarm intelligence algorithm based on artificial bee colony to generate reliable DNA sequence design. Traditional optimisation algorithm does not meet the heterogeneous and conflicting criteria for designing reliable DNA sequences. A multi-objective artificial bee colony algorithm was introduced to solve this problem. MO-ABC considers six different conflicting design criteria to generate reliable DNA sequences that can be used for biomolecular Computing. In addition, in order to verify the effectiveness of the improved multi-objective algorithm, it is compared with the well-known multi-target standard NSGA-II. After detailed research, the results show that our artificial bee colony intelligent algorithm has obtained a satisfactory and reliable DNA sequence. Specifically, Karaboga (2009) improved the artificial bee colony in 2009, including several multi-objective features, such as the concept of non-dominated solutions and the concept of non-dominated sorting (Deb and Goel, 2001). At ABC, the group of artificial bees contains three types of bees: hired bees, bystanders and scout bees. Hire bees to go to their food source and go back to the bee to dance here. The onlookers watch the dances of the hired bees and select the source of food based on them. Finally, Scouts explore new areas to find new sources of food. Since DNA sequence design is formulated as a multi-objective optimisation problem, a new multi-objective algorithm (MO-ABC) is defined. This new algorithm includes some of the techniques from the well-known multi-objective algorithms in the original ABC design. For example, the non-dominant ordering of the NSGA-II (Xu and Zhang, 2008) fast non-dominated sorting genetic algorithm is used. This sorting organises the population into different categories based on their dominance relationship (Deb et al., 2002). Other multi-objective concepts are taken from Pareto archived evolution strategy (PAES) (Knowles and Corne, 2012): Non-dominated solution files (NDS files). Based on the acceptance function, the file remains updated to find the best solution.

From the results of repeated simulation experiments, MO-ABC obtained better results than the classic NSGA-II algorithm. Globally, the MO-ABC algorithm seems to be very promising to generate reliable DNA sequences that can be applied to molecular Computing.

3.2.6 Membrane evolutionary algorithm

Xiao et al. (2012) used membrane evolution algorithms in 2011 to design reliable DNA code for DNA computation. P system is a computing model simulated biotechnology. It was founded in structure of biological cells. The emergence of a model based on the P system has led to the born of a new research direction called membrane computing, more details about P system can refer to Pang et al. (2018) and Song et al. (2018c, 2018a, 2018b);. Membrane computing is a new branch of natural computing, first proposed by Paun (2000). Membrane computing is a Bio-inspired algorithm (Song et al., 2016; Zhang et al., 2017). Membrane computing is a computational model abstracted from the function and structure of biological cells and tissues and organs composed of cells, with good distribution, parallelism, non-determinism and so on. Literature (Paun, 2018) details the latest research progress in membrane computing. In recent years, membrane computing has a widely application in robot (Perez-Hurtado et al., 2018), virtual network (Yu et al., 2018), database (Kumar et al., 2018) and solid waste transportation (He et al., 2015).

Although membrane Computing have achieved many good results, there are few membrane computing for the optimisation of DNA sequence code. Based on this, this paper constructs a DNA sequence design membrane system and proposes a string optimised membrane system DNA sequence code algorithm. The simulation results have achieved satisfactory results. This algorithm not only introduces new ideas and design algorithm for DNA code, but also enriches the application field of membrane computing.

3.2.7 Differential evolution optimisation algorithm

Differential evolution algorithm is an emerging evolutionary computing technology. It was proposed by Storn and Price (1997). The unique memory of differential evolution makes it possible to dynamically track the current search situation to adjust its search strategy, with strong global convergence and robustness. In Chaves-González et al.

(2014a) designed a DNA sequence for DNA computation using a multi-objective differential evolution algorithm. Based on the differential evolution algorithm, the paper introduces the Pareto tournament and proposes a new algorithm DEPT. The DEPT algorithm is based on the DE algorithm (Storn and Price, 1997), but it combines some multi-objective features, such as the Pareto tournament concept or non-dominated sorting (Zitzler and Thiele, 2001), to make it possible for true multi-objective optimisation.

The original differential evolution algorithm finds the optimal solution to a particular problem by maintaining a set of candidate solutions that are created by combining existing candidate solutions based on simple formulas of vector intersections and variations and then by using saved the update process saves solutions with the best fitness value and discards other solutions. One of the most important features of the original DE algorithm and the DEPT version is the mutation mechanism.

From the experimental results, the algorithm of this paper has obtained good results in the DNA sequence optimisation problem.

3.2.8 Other intelligent optimisation algorithms

Kurniawan et al. (2018) designed a thermodynamic-based DNA sequence using the ant colony system in 2008. An ant colony system was proposed to solve DNA sequence design problems. The algorithm uses a four-node model that uses the Watson-Crick base pair ΔG_37^0 closest to the thermodynamic parameters as the distance between the nodes. The results of this algorithm were compared with the genetic algorithm and obtained relatively good results. Ibrahim et al. (2009) optimised DNA sequences based on ant colony algorithm in 2009.

Chaves-Gonzales in his paper used a multi-target approach based on firefly behaviour in 2014 to generate reliable DNA sequences for molecular Computing. They expressed the problem as a multi-objective optimisation problem and we solved it with a new multi-target algorithm based on firefly behaviour. Specifically, our approach, the multi-target firefly algorithm (MO-FA), uses six combined constraints to measure the reliability of the generated sequence. In addition, in order to compare our multi-objective results, we also developed and adapted the famous fast non-dominated genetic algorithm (NSGA-II). The results show that the proposed algorithm has achieved very satisfactory results. In fact, the reliability of the resulting DNA sequences significantly exceeds the reliability of sequences obtained by other algorithm previously disclosed in the literature.

In 2012, Rao proposed a hybrid multi-objective heuristic algorithm for designing DNA sequences. In this paper, a multi-objective evolutionary algorithm is proposed, which combines the local search heuristic algorithm to design DNA sequences. The proposal is based on a multi-objective variant of the teaching-learning optimisation algorithm. The MOEA we chose is a multi-objective variant based on the teaching-learning optimisation (TLBO) approach, a very new group intelligent evolutionary algorithm (Rao et al., 2012). We call it MO-TLBO. In addition, we mix MO-TLBO with local search (LS) heuristics designed specifically for this problem. In this regard, it is well known that the LS algorithm helps metaheuristics find optimal solutions in a reasonable time period (Talbi et al., 2002). In order to test the accuracy of (H-MO-TLBO), this paper compares it with other multi-objective algorithms, such as: NSGA-II, fast non-dominated sorting genetic algorithm and SPEA2, intensity Pareto evolutionary algorithm (Laumanns, 2001).

4 Conclusions and future works

The rapid development of modern molecular biology technology has made DNA code problems popular among many scholars. The problem with DNA code is to design a set of DNA molecules that meet the combined constraints and thermodynamic constraints. The robustness of the code is better in the hybridisation reaction not only to avoid the formation of undesired secondary structures by the DNA itself, such as hairpin structure, but also to prevent non-specific hybridisation between DNA molecules and improve the accuracy of DNA Computing.

In this paper, the definition of DNA code, the problems of code and the constraints of code are introduced in detail. Two types of errors occur in the process of complementary pairing of DNA molecules. One is a false positive, i.e., a DNA molecule that is not fully complementary hybridises under appropriate conditions to form a double strand. The other is a false negative, i.e., a fully complementary DNA molecule does not hybridise as expected during the course of the reaction. Therefore, designing high quality DNA code that conforms to various constraints is critical to the correct conduct of the hybridisation reaction. This article has a detailed categorisation of constraints. It mainly includes three categories: combinatorial constraints, thermodynamic constraints and application-oriented constraints. The combinatorial constraints mainly include hamming distance, complement hamming distance, inverse Hamming distance, inverse complement hamming distance, H distance criterion, hairpin structure, continuity and fixed GC content. Thermodynamic constraints include melting temperature, free energy. The use of each of these constraints can improve the reliability of DNA code and each condition is based on the mandatory needs of DNA sequence technology. The use of the above conditions in the design of DNA sequences can make the code more biologically significant. The similarity is used to calculate the inverse hamming distance between two sequences; the H distance, Hamming distance is used to test the possibility of unintended DNA base pairing. The use of continuity allows the DNA code of the design to avoid secondary structure formation. These secondary structures are produced by the reaction of single-stranded DNA itself, including inner loops, raised loops and hairpins. Free energy, the energy required for a DNA double strand to become a single strand; Melting temperature, half of the DNA sequence in the duplex is in the state of the double helix and half of the dissociation state; GC content, percentage of guanine (G) and cytosine (C) in the DNA strand. Both free energy and melting temperature are conditions for controlling the relative stability of DNA duplexes. GC content is not very accurate, but it is easy to calculate. In this paper, the algorithm used in DNA code are classified and reviewed in detail.

The problem of DNA code can be divided into two aspects: quality optimisation and collection design. In this paper, the algorithm of quality optimisation is summarised in detail, including particle swarm optimisation algorithm, cultural evolution algorithm, ant colony algorithm, multi-target firefly behaviour algorithm, weed algorithm, differential evolution algorithm and so on. Designing a larger number of DNA code can be put into large-scale practical applications. Therefore, increasing the lower bound value of the DNA code set is a problem worth studying. Genetic algorithms so far, as well as improved algorithms based on genetic algorithms, are the main algorithm to improve the lower bound of code. The rest also include linear code algorithm, template mapping, squid algorithms, variable neighbourhood search algorithms, simulated annealing and more. Although DNA code has made some progress, DNA code has certain problems both in quantity and quality. These problems are also problems to be solved in the field of DNA code in the future. In the design of the number of code, when there are too many combinatorial constraints to be satisfied, it is difficult to design a sufficient number of DNA code for large-scale applications. When the DNA encode length is long, the intelligent optimisation algorithm runs for a long time. In code quality optimisation, when multi-objective optimisation is converted to single-objective optimisation, the selection of weight coefficients has not been unified. The quantity and quality of DNA code is always an inconsistent contradiction. The greater the number of DNA code, the larger the problem-solving scale, but the fewer combinatorial s of constraints are met and the quality of the code is not high. If the DNA quality is high, the constraints are met, but the number of code is difficult to guarantee. Therefore, designing a large number of DNA code and simultaneously encode the higher quality is a serious problem faced by DNA code and is also a problem to be solved by future researchers.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 61425002, 61751203, 61772100, 61702070, 61672121, 61572093, and 61802040), Program for Changjiang Scholars and Innovative Research Team in University (No. IRT_15R07), the Program for Liaoning Innovative Research Team in University (No. LT2015002), the Natural Science Foundation of Liaoning Province (No. 20180551241).

References

Aboluion, N., Smith, D.H. and Perkins, S. (2011) 'Linear and nonlinear constructions of DNA codes with Hamming distance math container loading mathjax, constant GC-content and a reverse-complement constraint', *Discrete Mathematics*, Vol. 312, No. 5, pp.1062–1075.

Aboluion, N.A. (2012) *The Construction of DNA Codes using A Computer Algebra System Contents*, Doctoral dissertation, University of Glamorgan, Scholars Press.

Adleman, L.M. (1994) 'Molecular computation of solutions to combinatorial problems', *Science*, Vol. 266, No. 5187, p.1021.

Arita, M. (2004) 'Writing information into DNA', *Aspects of Molecular Computing*, pp.23–35, Springer Berlin, Heidelberg.

Arita, M. and Kobayashi, S. (2001) 'The power of sequence design in DNA computing', *International Conference on Computational Intelligence and Multimedia Applications*, IEEE Computer Society, p.163.

Arita, M., Kobayashi, S. et al. (2002) 'DNA sequence design using templates', *New Generation Computing*, Vol. 20, No. 3, pp.263–277.

Arita, M., Nishikawa, A., Hagiya, M. et al. (2000) 'Improving sequence design for DNA computing', *Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., pp.875–882.

Ashlock, D. and Houghten, S.K. (2009) 'DNA error correcting codes: no crossover', *IEEE Conference on Computational Intelligence in Bioinformatics Computational Biology*, IEEE, pp.38–45.

Ashlock, D., Houghten, S.K., Brown, J.A. and Orth, J. (2012) 'On the synthesis of DNA error correcting codes', *Biosystems*, Vol. 110, No. 1, pp.1–8.

Banzhaf, W., Koza, J.R., Ryan, C., Spector, L. and Jacob, C. (2000) 'Genetic programming', *IEEE Intelligent Systems*, Vol. 15, No. 3, pp.74–84.

Bodo, P. (1997) 'The Atlantic salmon handbook: a compact guide to all aspects of fly fishing for the king of game fish', *Proceedings of the First European Conference on Artificial Life*, Vol. 142, No.12, pp.134–142.

Bomze, I.M., Budinich, M., Pardalos, P.M. and Pelillo, M. (1999) 'The maximum clique problem', *Handbook of Combinatorial Optimization*, Vol. 4, No. 3, pp.301–328.

Brouwer, A.E., Shearer, J.B., Sloane, N.J.A. and Smith, W.D. (1990) 'A new table of constant weight codes', *IEEE Transactions on Information Theory*, Vol. 36, No. 6, pp.1334–1380.

Cervantes-Salido, V.M., Jaime, O., Brizuela, C.A. and Martínez-Pérez, I.M. (2013) 'Improving the design of sequences for DNA computing: a multiobjective evolutionary approach', *Applied Soft Computing*, Vol. 13, No. 12, pp.4594–4607.

Chaves-González, J.M. and Vega-Rodríguez, M.A. (2014a) 'A multiobjective approach based on the behavior of fireflies to generate reliable DNA sequences for molecular computing', *Applied Mathematics Computation*, Vol. 227, No. 2, pp.291–308.

Chaves-González, J.M. and Vega-Rodríguez, M.A. (2014b) 'DNA strand generation for DNA computing by using a multi-objective differential evolution algorithm', *Biosystems*, Vol. 116, No. 3, pp.49–64.

Chaves-González, J.M., Vega-Rodríguez, M.A., Granado-Criado, J.M. (2013) 'A multiobjective swarm intelligence approach based on artificial bee colony for reliable DNA sequence design', *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 9, pp.2045–2057.

Chee, Y.M. and Ling, S. (2008) 'Improved lower bounds for constant GC-content DNA codes', *IEEE Trans. Inform. Theory*, Vol. 54, No. 1, pp.391–394.

Chidchob, P. and Sleiman, H.F. (2018) 'Recent advances in DNA nanotechnology', *Current Opinion in Chemical Biology*, Vol. 46, No. 10, pp.63–70.

Colorni, A. (1992) 'Distributed optimization by ant colonies', *European Conference on Artificial Life*, The MIT Press, New York.

Cui, G., Niu, Y., Wang, Y. et al. (2007) 'A new approach based on PSO algorithm to find good computational encoding sequences', *Progress in Natural Science: Materials International*, Vol. 17, No. 6, pp.712–716.

Cukras, A.R., Faulhammer, D., Lipton, R.J. et al. (1999) 'Chess games: a model for RNA based computation', *Biosystems*, Vol. 52, Nos. 1–3, pp.35–45.

Dan, T., Davey, M. and Laflamme, M. (2012) 'qRT-PCR for validating microbial microarray data', *Quantitative Real-time PCR in Applied Microbiology*, Vol. 2012, No. 2012, pp.1–80.

Dan, T., Smith, D.H. and Montemanni, R. (2014) 'Thermodynamic post-processing versus GC-content pre-processing for DNA codes satisfying the hamming distance and reverse-complement constraints', *IEEE ACM Transactions on Computational Biology Bioinformatics*, Vol. 11, No. 2, pp.441–452.

Deaton, R. (Ed.) (1996) 'Genetic search of reliable encodings for DNA based computation', *Late-Breaking Papers at the First Conference on Genetic Programming*.

Deaton, R. and Garzon, M. (1998) 'Thermodynamic constraints on DNA-based computing', in Păun, G. (Ed.): *Computing with Bio-Molecules*, pp.138–152.

Deaton, R., Franceschetti, D.R., Garzon, M., Rose, J.A., Murphy Jr., R.C. (Eds.) (1997) 'Information transfer through hybridization reactions in DNA based computing genetic programming', *Proceedings of the Second Conference*, Stanford University, 13–16 July.

Deb, K. (2000) 'A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-2', *Lecture Notes in Computer Science*, Vol. 1917, No. 2000, pp.849–858.

Deb, K. and Goel, T. (2001) 'Controlled elitist non-dominated sorting genetic algorithms for better convergence', *International Conference on Evolutionary Multi-Criterion Optimization*, Springer-Verlag, pp.67–81.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.

Dineen, S. (2011) 'Thermodynamically stable RNA three-way junction for constructing multifunctional nanoparticles for delivery of therapeutics', *Nature Nanotechnology*, Vol. 6, No. 10, pp.658–667.

Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*, Thesis Politecnico, Di Milano Italy.

Dorigo, M., Birattari, M. and Stutzle, T. (2007) 'Ant colony optimization', *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp.28–39.

D'Yachkov, A.G., Erdös, P.L., Macula, A.J. et al. (2003) 'Exordium for DNA codes', *Journal of Combinatorial Optimization*, Vol. 7, No. 4, pp.369–379.

Eusuff, M.M. and Lansey, K.E. (2003) 'Optimization of water distribution network design using the shuffled frog leaping algorithm', *Journal of Water Resources Planning Management*, Vol. 129, No. 3, pp.210–225.

Faulhammer, D., Cukras, A.R., Lipton, R.J. et al. (2000) 'Molecular computation: RNA solutions to chess problems', *Proceedings of the National Academy of Sciences*, Vol. 97, No. 4, pp.1385–1389.

Frutos, A.G., And, L.M.S. and Corn, R.M. (1998) 'Enzymatic ligation reactions of DNA 'words' on surfaces for DNA computing', *Journal of the American Chemical Society*, Vol. 120, No. 40, pp.10277–10282.

Frutos, A.G., Liu, Q., Thiel, A.J. et al. (1997) 'Demonstration of a word design strategy for DNA computing on surfaces', *Nucleic Acids Research*, Vol. 25, No. 23, pp.4748–4757.

Gaborit, P. and King, O.D. (2004) 'Linear constructions for DNA codes', *Theoretical Computer Science*, Vol. 334, No. 1, pp.99–113.

Garzon, M. (1997) 'A new metric for DNA computing', *Proceedings of the 2nd Genetic Programming Conference*, Vol. 32, No. 1, pp.636–638.

Garzon, M., Deaton, R., Neathery, P., Murphy, R., Franceschetti, D. and Stevens, S. (1997) 'On the encoding problem for DNA computing', *The Third DIMACS Workshop on DNA-Based Computing*, Vol. 1997, No. 1997, pp.230–237.

Garzon, M.H. and Deaton, R.J. (2004) 'Codeword design and information encoding in DNA ensembles', *Natural Computing*, Vol. 3, No. 3, pp.253–292.

Garzon, M.H., Phan, V., Roy, S. et al. (2006) 'In search of optimal codes for DNA computing', *International Conference on DNA Computing*, Springer-Verlag, pp.143–156.

Graves, D.J. (1999) 'Powerful tools for genetic analysis come of age', *Trends in Biotechnology*, Vol. 17, No. 3, pp.127–134.

Guo, Q., Wang, B., Zhou, C. et al. (2017) 'DNA code design based on the Bloch quantum chaos algorithm', *IEEE Access*, No. 99, No. 5, pp.1–1.

Hansen, P. and Mladenović, N. (2007) 'Variable neighborhood search: principles and applications', *European Journal of Operational Research*, Vol. 130, No. 3, pp.449–467.

Hartemink, A.J., Gifford, D.K. and Khodor, J. (1999) 'Automated constraint-based nucleotide sequence selection for DNA computation', *Biosystems*, Vol. 52, Nos. 1–3, pp.227–235.

He, J., Xiao, J., Liu, X., Wu, T. and Song T. (2015) 'A novel membrane-inspired algorithm for optimizing solid waste transportation', *Optik.*, Vol. 126, No. 23, pp.3883–3888.

Head, T. (1987) 'Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors', *Bulletin of Mathematical Biology*, Vol. 49, No. 6, p.737.

Hong, H., Wang, L., Ahmad, H., Li, J., Yang, Y. and Wu, C. (2016) 'Construction of DNA codes by using algebraic number theory', *Finite Fields Their Applications*, Vol. 37, No. C, pp.328–343.

Hybridization, D., Hartemink, E.J. and Gifford, D.K. (1997) 'Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation', *DNA Based Computers II*, Vol. 1997, No. 1997, pp.25–39.

Ibrahim, Z., Kurniawan, T.B., Khalid, N.K., Sudin, S. and Khalid, M. (2009) 'Implementation of an ant colony system for DNA sequence optimization', A*rtificial Life Robotics*, Vol. 14, No. 2, pp.293–296.

Jin, X. (2014) 'Forthcoming era of biological computer', *Bulletin of Chinese Academy of Sciences*, Vol. 29, No. 1, pp.42–53.

Jong, K.A.D. (1975) *Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan.

Karaboga, D. (2009) 'Survey A: algorithms simulating bee swarm intelligence', *Artificial Intelligence Review*, Vol. 31, No. 1, pp.61–85.

Kawashimo, S., Ono, H., Sadakane, K. and Yamashita, M. (2006a) *DNA Sequence Design by Dynamic Neighborhood Searches*, *IEICE Technical Report*, Vol. 106, pp.157–171.

Kawashimo, S., Ono, H., Sadakane, K. et al. (2006b) 'DNA sequence design by dynamic neighborhood searches', *International Workshop on DNA-Based Computers*, Springer, Berlin, Heidelberg, pp.157–171.

Kenned, Y.J. and Eberhart, R.C. (1995) 'Particle swarm optimization', in *Proceedings of the IEEE International Conference on Neural Networks*, IEEE, Piscataway, pp.1942–1948.

Khalid, N.K., Ibrahim, Z., Kurniawan, T.B. et al. (2009) 'Implementation of binary particle swarm optimization for DNA sequence design', *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing and Ambient Assisted Living*, Springer Berlin Heidelberg, pp.450–457.

Kirkpatrick, S., Gelatt Jr., C. and Vecchi, M.P. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, p.671.

Knowles, J and Corne, D. (Eds.) (2012) 'The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation', *Proc Congress on Evolutionary Computation*.

Kumar, D., Verma, N.K. and Singh, N. (2018) 'A review paper on deducting database in membrane computing', *Journal of Statistics and Management Systems*, Vol. 21, No. 4, pp.667–673.

Kurniawan, T.B., Khalid, N.K., Ibrahim, Z. et al. (2008) 'An ant colony system for DNA sequence design based on thermodynamics', *Iasted International Conference on Advances in Computer Science Technology*, pp.144–149.

Laumanns, M. (2001) *SPEA2 : Improving the Strength Pareto Evolutionary Algorithm*, Technical Report Gloriastrasse.

Limbachiya, D. and Gupta, M.K. (2015) 'Natural data storage: a review on sending information from now to then via nature', *Computer Science*, arXiv preprint arXiv:1505.04890.

Limbachiya, D., Rao, B. and Gupta, M.K. (2016) 'The art of DNA strings: sixteen years of DNA code theory', *arXiv: Information Theory*.

Lipton, R.J. (1995) 'DNA solution of hard computational problems', *Science*, Vol. 268, No. 5210, pp.542–545.

Liu, W., Wang, S., Gao, L., Zhang, F. and Xu, J. (2003) 'DNA sequence design based on template strategy', *Journal of Chemical Information Computer Sciences*, Vol. 43, No. 6, p.2014.

Ma, K., Gong, Y., Aubert, T. et al. (2018) 'Self-assembly of highly symmetrical, ultrasmall inorganic cages directed by surfactant micelles', *Nature*, Vol. 558, No. 7711, pp.577–580.

Macwilliams, F.J. and Sloane, N.J.A. (1997) *The Theory of Error-Correcting Codes: North-Holland Pub. Co.*, pp.185–186.

Masubuchi, T., Endo, M., Iizuka R. et al. (2018) 'Construction of integrated gene logic-chip', *Nature Nanotechnology*, Vol. 13, No. 10, pp.933–940.

Mehrabian, A.R. and Lucas, C. (2006) 'A novel numerical optimization algorithm inspired from weed colonization', *Ecological Informatics*, Vol. 1, No. 4, pp.355–366.

Montemanni, R. and Smith, D.H. (2008) 'Construction of constant GC-content DNA codes via a variable neighbourhood search algorithm', *Journal of Mathematical Modelling Algorithms*, Vol. 7, No. 3, p.311.

Montemanni, R. and Smith, D.H. (2009a) 'Heuristic algorithms for constructing binary constant weight codes', *IEEE Transactions on Information Theory*, Vol. 55, No. 10, pp.4651–4656.

Montemanni, R. and Smith, D.H. (2009b) 'Heuristic construction of constant weight binary codes', *Computer Sciences*, Vol.55, No. 10, pp.4651–4656.

Montemanni, R., Smith, D.H. and Koul, N. (2009) 'Three metaheuristics for the construction of constant GC-content DNA codes', *International Conference on Applied Operational Research*, Vol. 6, No. 6, pp.167–175.

Narayanan, A. and Moore, M. (1996) 'Quantum-inspired genetic algorithms', *IEEE International Conference on Evolutionary Computation*, IEEE, pp.61–66.

Orth, J. and Houghten, S. (2011) 'Optimizing the Salmon algorithm for the construction of DNA error-correcting codes', *IEEE Symposium on Computational Intelligence in Bioinformatics Computational Biology*, IEEE, pp.1–7.

Pang, S., Ding, T., Rodríguez-Patón, A., Song, T. and Phen, Z. (2018) 'A parallel bioinspired framework for numerical calculations using enzymatic P system with an enzymatic environment', *IEEE Access*, Vol. 6, No. 2018, pp.65548–65556.

Paun, G. (2000) 'Computing with membranes', *Journal of Computer System Sciences*, Vol. 61, No. 1, pp.108–143.

Paun, G. (2018) 'A dozen of research topics in membrane computing', *Theoretical Computer Science*, Vol. 736, No. 2018, pp.76–78.

Penchovsky, R. and Ackermann, J. (2003) 'DNA library design for molecular computation', *Journal of Computational Biology*, Vol. 10, No. 2, pp.215–229.

Peng, X., Zheng, X., Wang, B. et al. (2016) 'A micro-genetic algorithm for DNA encoding sequences design', *International Conference on Control Science and Systems Engineering*, IEEE, pp.10–14.

Perez-Hurtado, I., Pérez-Jiménez, M.J., Zhang, G. et al. (2018) 'Robot path planning using rapidly-exploring random trees: s membrane computing approach', *7th International Conference on Computers Communications and Control (ICCCC)*, IEEE, pp.37–46.

Pirrung, M.C. (2002) 'How to make a DNA chip', *Angewandte Chemie International Edition*, Vol. 41, No. 8, pp.1276–1289.

Qiu, M., Khisamutdinov, E., Zhao, Z. et al. (2013) 'RNA nanotechnology for computer design and in vivo computation', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 371, No. 2000, pp.20120310–20120310.

Rao, R.V., Savsani, V.J. and Vakharia, D.P. (2012) 'Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems', *Information Sciences*, Vol. 183, No. 1, pp.1–15.

Rocha, A.S.L.D., Faria, L.C.B.D., Kleinschmidt, J.H., Silva-Filho, M.C. (2010) 'DNA sequences generated by Z(4)-linear codes', Vol. 2010, No. 2010, pp.1320–1324.

Rose, T.M., Henikoff, J.G. and Henikoff, S. (2003) 'COnsensus-DEgenerate yybrid oligonucleotide primer (CODEHOP) PCR primer design', *Nucleic Acids Research*, Vol. 31, No. 13, pp.3763–3766.

Rothemund, P.W. (2006) 'Folding DNA to create nanoscale shapes and patterns', *Nature*, Vol. 440, No. 7082, p.297.

Santalucia, J. (1998) 'A unified view of polymer, dumbbell and oligonucleotide DNA nearest-neighbor thermodynamics', *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 95, No. 4, pp.1460–1465.

Seeman, N.C. (1998) 'DNA nanotechnology: novel DNA constructions', *Annual Review of Biophysics Biomolecular Structure*, Vol. 27, No. 1, p.225.

Shin, S.Y., Lee, I.H., Kim, D. and Zhang, B.T. (2005) 'Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing', *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 2, pp.143–158.

Song, T., Pang, S., Hao, S., Rodríguez-Patón, A. and Zheng, P. (2018) 'A parallel image skeletonizing method using spiking neural P systems with weights', *Neural Processing Letters*, pp.1–12.

Song, T., Rodr, A., Zheng, P. et al. (2018) 'Spiking neural P systems with colored spikes', *IEEE Transactions on Cognitive and Developmental Systems*, pp.1–1.

Song, T., Zeng, X., Zheng, P., Jiang, M. and Rodríguez-Patón, A. (2018) 'A parallel workflow pattern modeling using spiking neural P systems with colored spikes', *IEEE Transactions on NanoBioscience*, Vol. 17, No. 4, pp.474–484.

Song, T., Zheng, P. and Wong, D. (2016) Bio-Inspired Computing Model and Algorithm. 2016.

Storn, R. and Price, K. (1997) 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *Journal of Global Optimization*, Vol. 11, No. 4, pp.341–359.

Talbi, E.G. (2002) 'A taxonomy of hybrid metaheuristics', *Journal of Heuristics*, Vol. 8, No. 5, pp.541–564.

Tanaka, F., Nakatsugawa, M., Yamamoto, M. et al. (Eds.) (2001) 'Developing support system for sequence design in DNA computing', *DNA Computing, International Workshop on DNA-Based Computers*, Tampa, Florida, USA, 10–13 June.

Tanaka, F., Nakatsugawa, M., Yamamoto, M., Shiba, T. and Ohuchi, A. (Eds.) (2002) 'Towards a general-purpose sequence design system in DNA computing', *Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02*.

Tulpan, D., Andronescu, M., Chang, S.B., Shortreed, M.R., Condon, A., Hoos, H.H. et al. (2005) 'Thermodynamically based DNA strand design', *Nucleic Acids Research*, Vol. 33, No. 15, pp.4951–4964.

Tulpan, D.C. (2006) *Effective Heuristic Methods for DNA Strand Design*, Doctoral dissertation.

Tulpan, D.C. and Hoos, H.H. (2003) 'Hybrid randomised neighbourhoods improve stochastic local search for DNA code design', *Advances in Artificial Intelligence*, pp.418–433, Springer Berlin Heidelberg.

Tulpan, D.C., Ghiggi, A. and Montemanni, R. (2013) 'Computational sequence design techniques for DNA microarray technologies', *Systemic Approaches in Bioinformatics and Computational Systems Biology: Recent Advances*.

Tulpan, D.C., Hoos, H.H. and Condon, A. (2002) 'Stochastic local search algorithms for DNA word design', *Lecture Notes in Computer Science*, Vol. 2568, No. 9, pp.229–241.

Varbanov, Z., Todorov, T. and Hristova, M. (2015) 'A method for constructing DNA codes from additive self-dual codes over GF', *Proc. CAIM Conference*, Romania, Vol. 40, pp.203–211.

Wang, B., Xie, Y., Zhou, S., Zheng, X. and Zhou. C. (2018) 'Correcting errors in image encryption based on DNA coding', *Molecules*, Vol. 23, No. 8, pp.60–70, Basel, Switzerland.

Wang, B., Zheng, X.D., Zhou, S.H., Zhou, C.J., Wei, X.P., Zhang, Q. et al. (2018) 'Constructing DNA barcode sets based on particle swarm optimization', *IEEE-ACM Trans Comput. Biol. Bioinform.*, Vol. 15, No. 3, pp.999–1002.

Winfree, E. (1998) *Algorithmic Self-Assembly Of DNA*, California Institute of Technology.

Winfree, E., Liu, F., Wenzler, L.A. et al. (1998) 'Design and self-assembly of two-dimensional DNA crystals', *Nature*, Vol. 394, No. 6693, pp.539–544.

Xiao, J., Xu, J., Chen, Z., Zhang, K. and Pan, L. (2009) 'A hybrid quantum chaotic swarm evolutionary algorithm for DNA encoding', *Computers Mathematics with Applications*, Vol. 57, Nos. 11–12, pp.1949–1958.

Xiao, J.H., Jiang, Y., He, J.J. and Cheng, Z. (2013) 'A dynamic membrane evolutionary algorithm for solving DNA sequences design with minimum free energy', *Match Communications in Mathematical in Computer Chemistry*, Vol. 70, No. 3, pp.987–1004.

Xiao, J.H., Zhang, X.Y. and Xu, J. (2012) 'A membrane evolutionary algorithm for DNA sequence design in DNA computing', *Science Bulletin*, Vol. 57, No. 6, pp.698–706.

Xu, C., Zhang, Q., Wang, B. et al. (2008) 'Research on the DNA sequence design based on GA/PSO algorithms.', *The International Conference on Bioinformatics and Biomedical Engineering*, IEEE, pp.816–819.

Xu, S. and Zhang, Q. (2008) 'Optimization of DNA coding based on GA/PSO algorithm', *Computer Engineering*, Vol. 34, No. 1, pp.218–220.

Yu, C., Lian, Q., Zhang, D. et al. (2018) 'PAME: evolutionary membrane computing for virtual network embedding', *Journal of Parallel and Distributed Computing*, Vol. 111, No. 2018, pp.136–151.

Zhang, B.T. (1998) 'Molecular algorithms for efficient and reliable DNA computing', *Issues in Supply Chain Scheduling Contracting*, Vol. 1998, No. 1998, pp.1–4.

Zhang, X., Aradas, A.R.P., Zeng, X. et al. (2017) 'Theory and application of bio-inspired intelligence and methods', Journal of Universal Computer Science, Special Issue, Vol. 23, No. 7, pp.586–588.

Zitzler, E. and Thiele, L. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*, pp.75–96, John Wiley Sons, Inc, New York, NY, USA.

Zong, W.G., Kim, J.H. and Loganathan, G.V. (2011) 'A new heuristic optimization algorithm: harmony search', *Simulation*, Vol. 2, No. 2, pp.60–68.