Minimum Free Energy Coding for DNA Storage

Ben Cao, Xiaokang Zhang, Jieqiong Wu, Bin Wang, Qiang Zhang, and Xiaopeng Wei

***Abstract*— With the development of information technology, huge amounts of data are produced at the same time. How to store data efficiently and at low cost has become an urgent problem. DNA is a high-density and persistent medium, making DNA storage a viable solution. In a DNA data storage system, the first consideration is how to encode the data effectively into code words. However, DNA strands are prone to non-specific hybridization during the hybridization reaction process and are prone to errors during synthesis and sequencing. In order to reduce the error rate, a thermodynamic minimum free energy (MFE) constraint is proposed and applied to the construction of coding sets for DNA storage. The Brownian multi-verse optimizer (BMVO) algorithm, based on the Multi-verse optimizer (MVO) algorithm, incorporates the idea of Brownian motion and Nelder–Mead method, and it is used to design a better DNA storage coding set. In addition, compared with previous works, the coding set has been increasing by 4%–50% in size and has better thermodynamic properties. With the improvement of the quality of the DNA coding set, the accuracy of reading and writing and the robustness of the DNA storage system are also enhanced.**

***Index Terms*— DNA storage, DNA coding design, BMVO algorithm, minimum free energy.**

I. Introduction

SINCE the "Industry 4.0" era, information technology has promoted the development of industries and produced massive amounts of data. International Data Corporation (IDC) once reported that due to the digital information explosion, the total amount of data in 2025 will reach 175 ZB. Massive data bring both convenience and challenges to people. How to use and store data reasonably has become an urgent problem to be solved. Compared with traditional electronic storage media, DNA, as a storage medium with high density, high durability and long-term stability, has become a viable solution. DNA consists of four nitrogenous bases (AGCT), and its theoretical storage density is twice that of binary storage. Under suitable conditions, DNA can be stored for tens of thousands of years [1]. At one point in time, DNA synthesis and sequencing technology was a major obstacle to the development of DNA storage, but with the enhancement of DNA synthesis and third-generation sequencing technology, the cost of a single base is now as low as a few cents. Neiman *et al.* began to pay attention to related issues in DNA storage as early as 1964 [2], and over time more and more researchers have begun paying attention to DNA storage technology.

Due to its high storage density, DNA has great potential as a storage medium. The process of DNA storage is the process of encoding and decoding the DNA sequence. The accessibility of DNA-based data storage is mainly driven by two technologies: DNA synthesis for "encoding" and DNA sequencing for "decoding." Normally, the information is first converted into an ATCG base sequence using a preset coding scheme. These sequences are synthesized into oligonucleotides or long DNA fragments, which are decoded into 0-1 codes when they need to be read. One of the earliest applications of DNA storage occurred in 1988, when Joe Davis collaborated with researchers at Harvard University. The research group stored a Germanic picture of life and female earth in the DNA sequence of E. coli and showed that through decoding the picture could be restored [3]. At the beginning of this century, Bancroft *et al.* [4] proposed a simple method of coding using codon triplets, demonstrating the great potential of DNA as a storage medium. Bornholt *et al.* [5] proved through experiments that the DNA storage pool can be accessed continuously and also demonstrated the long-term effectiveness of DNA as an archive storage medium. Nguyen *et al.* [6] published a research report on DNA storage in plasmids. They used the Perl script to encode the DNA sequence of a 2046-word document and synthesized the encoded DNA sequence for information storage. The results showed that plasmid DNA data storage can be used for long-term information storage and can restore data sources in a repeatable and reliable manner. Tomek *et al.* [7] used chemical handles to selectively extract unique files from a complex DNA database with 5 TB of data and designed and implemented a nested file address system that increased the theoretical maximum capacity of the DNA storage system. Chen *et al.* [8] used silica to increase the

Manuscript received July 26, 2020; revised December 17, 2020; accepted January 28, 2021. Date of publication February 3, 2021; date of current version April 1, 2021. This work was supported in part by the National Key Technology Research and Development Program of China under Grant 2018YFC0910500; in part by the National Natural Science Foundation of China under Grant 61425002, Grant 61751203, Grant 61772100, Grant 61972266, Grant 61802040, and Grant 61672121; in part by the High-level Talent Innovation Support Program of Dalian City under Grant 2017RQ060 and Grant 2018RQ75; and in part by the Innovation and Entrepreneurship Team of Dalian University under Grant XQN202008. *(Corresponding authors: Bin Wang; Qiang Zhang; Xiaopeng Wei.)*

Ben Cao, Xiaokang Zhang, Jieqiong Wu, and Bin Wang are with the Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China (e-mail: bencaocs@gmail.com; xiaokangz96@gmail.com; juneqiongqiong@gmail.com; wangbinpaper@gmail.com).

Qiang Zhang is with the Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China, and also with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: zhangq@dlut.edu.cn).

Xiaopeng Wei is with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: xpwei@dlu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNB.2021.3056351, provided by the authors.

Digital Object Identifier 10.1109/TNB.2021.3056351

density of long-term DNA storage. The weight of DNA in the silica spheres used for DNA storage was increased to 3.4 wt%. Compared with previous DNA storage technology, the silica spheres exhibited superior DNA storage density and stability. Once carbon nanotubes and silicon dioxide were created, they were used by Zhang *et al.* [9] to improve the performance of DNA storage. By condensing DNA strands on the surface of one-dimensional carbon nanotubes (CNTs), a new type of storage medium called tube nucleic acids (TNAs) was created. This non-standard, DNA-free hybridization strategy provides a new approach to DNA-based data storage.

In terms of the quantitative analysis of DNA storage processes, Wang *et al.* [10] used the most advanced droplet digital PCR (ddPCR) technology to monitor the long-term storage of DNA, proving that ddPCR is an ideal method for detecting DNA storage. DNA synthesis, sequencing, storage and processing will produce errors, and error-free information storage presents a challenge. Meiser *et al.* [11] released a protocol including technical details and computer code for converting digital information into DNA sequences, processing biomolecules, storing biomolecules and then regaining the information through DNA sequencing. This protocol enhances the amount of DNA storage and random access to data and allows for compatibility with future sorting and synthesis technologies. Recently, researchers have also considered the use of DNA storage to protect information security. Grass *et al.* [12] reported a strategy to use personal genetic information to store valuable information in synthetic DNA, protected by personalized keys.

As mentioned above, one of the two most important processes in DNA storage is DNA coding. DNA codes (sets of words of fixed length n over the alphabets {A, C, G, T}) that meets specific combinational constraints is designed to synthesize DNA strands for reliable storage and retrieval of information. These codes can be used specifically for DNA computation or DNA storage. Non-biological data is encoded into DNA coding that can be used to store information at the molecular level. Efficient and stable encoding of DNA can not only increase storage capacity and maintain data integrity but also reduce the error rate of DNA in storage. In the process of DNA synthesis and sequencing, errors such as non-specific hybridization, replacement and deletion are likely to occur. Reasonable coding has an irreplaceable effect on the robustness of the DNA storage system and the repeated reading and writing of data, so building a reasonable and efficient DNA storage coding set has become the focus of study.

The problem of combination constraints in DNA coding was first proposed by Garzon *et al.* [13]. The main purpose of DNA coding is to design efficient and stable DNA sequences and avoid non-specific hybridization and other errors during the reaction. In 2012, Church *et al.* [14] proposed a scheme to deal with errors caused by DNA sequencing and synthesis, such as repetitive sequences, secondary structure and abnormal GC-content. Huffman coding is generally considered to be an optimal coding method and is often used for lossless data compression. In 2013, Goldman *et al.* [15] used Huffman coding in the coding scheme, effectively increasing the coding potential to 1.58 bits/nt. In 2015, Grass *et al.* [16] proposed a method based on Galois field (GF) and Reed-Solomon (RS) code to improve the ability to detect and correct coding errors, which proved that the potential for data density is 1.78 bits/nt. In 2016, Bornholt *et al.* [17] used the exclusive OR (XOR) coding principle to improve Goldman's coding scheme, using XOR operations to generate redundancy. For any two sequences (AB, AC or BC), the third sequence can be easily restored, thus increasing redundancy flexibility. Fountain code is also an information coding method widely used in communication systems and is famous for its robustness and high efficiency. In 2017, Erlich and Zielinski [18] used fountain codes in their coding schemes. In their report, the verification steps can prevent single-nucleotide duplication and abnormal GC content. Song *et al.* [19] proposed a coding method that converts 0-1 sequences into DNA base sequences and achieves 1.9 bits/nt with low encoding/decoding complexity and limited error propagation. Immink *et al.* [20] converted the standard binary maximum run length limit sequence into a maximum run length limit q-ary sequence to avoid the appearance of long homopolymers and constructed a code rate close to the theoretical maximum. The Mutually Uncorrelated (WMU) code was used by Yazdi *et al.* [21] for primer design DNA storage coding. The DNA sequence that satisfies WMU requires a large Hamming distance between each other. The resulting sequence has a balanced symbol composition and avoids primer-dimer byproducts. WMU coding maintains the integrity of the data, which can reduce the error rate and improve the error correction ability. Organick *et al.* [22] proposed an end-to-end DNA data storage strategy that demonstrated the ability of large-scale random access and was capable of correcting errors caused by insertion and deletion. Song and Zeng [23] proposed a novel three-base block encoding scheme (SED3B) for reliable and orthogonal information encoding of living cells. SED3B provides effective error correction by adding an error detection base to the small data block and combining the inherent redundancy of DNA molecules. And through error-prone PCR experiments in E. coli cells, it was confirmed that the error rate of 19% can be corrected. Wang *et al.* [24] proposed a novel content-balanced run-length limited (C-RLL), which has an efficient coding construction method and can simultaneously generate short DNA sequences that satisfy the maximum homopolymer run limit and balanced GC-content limit. Lee *et al.* [25] described a de novo synthesis strategy for DNA data storage that utilizes template-independent polymerase terminal deoxynucleotide transferase (TdT) under motion control conditions. This strategy synthesizes a DNA chain containing 144 bits and demonstrates the flow-type nano sequencing search, including addressing, which provides a method and theoretical basis for the development of DNA digital information storage technology. Fei *et al.* [26] designed a turbo-like decoder binary LDPC code DNA storage channel. Simulation results showed that the binary LDPC code has a similar error rate, but its speed is four times faster than the quaternary code. Since data generation speeds far exceed the increase in the storage density of media such as hard drives and tapes, researchers have begun to study new architectures and media types that can store "cold," infrequently accessed data at a very low cost.

For instance, OligoArchive, proposed by Appuswamy *et al.* [27], is an architecture that uses a DNA-based storage system as a relational database archiving layer. Through experimentation, Appuswamy proved that by constructing archive and recovery tools, OligoArchive can be implemented in practice, and even SQL queries can be used on OligoArchive.

In order to design a higher-quality DNA storage coding set, it is important to have one or a set of constraints to reasonably constrain and evaluate the coding set. Conventional constraints include GC-content constraints and homopolymer-related constraints. The essence of any biochemical reaction is the change of energy, so thermodynamic constraints can better reflect the state and stability of the DNA strand compared to combination constraints. For the above reasons, we design a minimum free energy constraint based on thermodynamics. The design problem of DNA coding sets may be computationally difficult because the size of the solution space is too large, so we consider using heuristic algorithms to design DNA storage coding sets. In this paper, we propose a new algorithm, namely Brownian multi-verse optimizer (BMVO), which uses Brownian motion and the Nelder–Mead method to improve the MVO algorithm. The BMVO algorithm passes the rank-sum test and achieves ideal results on 13 different types of benchmark functions. Among them, two test functions have found the global optimum, and there are obvious improvements in other test functions. In the construction of DNA coding sets, comparison with previous results also demonstrates the BMVO algorithm's optimized performance and ability to solve practical problems.

The organizational structure of this paper is as follows. Section 2 describes the constraints of coding sets in DNA storage. Section 3 introduces the mechanism of the BMVO algorithm and the improvements provided by using Brownian motion and the Nelder–Mead method. Section 4 includes the results and analysis of the Benchmark functions, the analysis and comparison of the final DNA coding set and the final discussion. Section 5 summarizes the study and the outlook for future work.

II. Constraint

A. Minimum Free Energy (MFE) Constraint

Energy change is the essential change of any biochemical reaction, and the Gibbs standard free energy is one measure of energy change. The minimum free energy (MFE) of one or more sequences is the minimum value of Gibbs standard free energy of all possible secondary structures. The secondary structure with small Gibbs standard free energy is more stable than large Gibbs standard free energy [28]. Therefore, the Gibbs standard free energy value can be used to measure the quality of DNA sequences. MFE constraints can help select sequences by measuring the strength of the bonds formed between each sequence and the degree of perfect matching while ignoring the number of matches or the number of individual nucleotides involved in bonding [29].

The variable *E(s,s')* denotes the value of MFE between two DNA codes s and s', which can be calculated by PairFold [30]. PairFold is the tool to predict sub-optimal secondary structures of two interacting strands by Mirela *et al.* PairFold predicts the MFE secondary structure that can be formed by two interacting nucleic acid molecules. In this work PairFold was used to calculate the minimum free energy between two single strands of DNA. Here, s' is the sequence which complements s by replacing each A in s by T and vice versa, and by replacing each G in s by C and vice versa. The minimum free energy constraint considered between s and s 'can measure the stability of THE DNA double strand, and a more stable DNA double strand means greater storage durability under suitable conditions. Given the threshold parameter t, the following constraints are given based on MFE: for all pairs of s in S, $E(s,s') \leq t$, namely $E(S) = max\ s \in S[E(s,s')] \leq t$. Parameter t can be a constant or have other values. In this paper, t is adaptively generated by (1) according to the attributes of the current encoding set:

$$t = \frac{\sum_{i=1}^{n} \Delta E(s_i, s_i')}{n}, \tag{1}$$

where n is the size of the candidate set.

B. Storage Edit Distance Constraint

Edit distance is often used in information theory to measure the degree of dissimilarity between two sequences. The edit distance is the sum of the single-character operands of one sequence to another by inserting, deleting and substituting [31]. Editing distance is also frequently used in natural language processing and bioinformatics, such as determining the similarity of ATCG between two DNA sequences.

In DNA storage, storage edit distance is used to reduce the error rate in DNA synthesis and nano sequencing. The storage edit distance is defined as follows. For the DNA code words a and b of length n, $d\_E(a, b)$ is defined as the storage edit distance between a and b. $SE(a_i)$ defines the minimum $d\_E(a_i, b_j)$ in all DNA coding sets, which should not be greater than the element d. The specific expression is (2):

$$\mathrm{SE}(\mathrm{a}_i) = \min_{1 \leq j \leq n, j \neq i} \{d\_E(a_i, b_j)\} \geq d. \tag{2}$$

C. GC-Content Constraint

GC content usually refers to the ratio of bases G and C to the four bases in a DNA sequence [32]. GC content is an important index in DNA synthesis and sequencing. In DNA synthesis, each fragment of GC content must be in a specific range. Depending on the synthesis method, the oligos are assembled using ligase or polymerase. In oligos with high GC content [33], adjacent guanines tend to form more hydrogen bonds, leading to inter-chain and intra-chain folding. In order to assemble the oligomer into larger fragments, the melting temperature and GC-content should only have a slight deviation between the oligomers, so the designed DNA sequence should be homogeneous in terms of GC-content. The GC-content is also important during the denaturation phase of PCR. High GC-content will lead to high melting temperature, which will hinder the separation of chains and thus reduce the yield of the PCR process.

Generally, GC-content of about 50% in a DNA sequence is defined as stable. As shown in the formula, the GC-content

of a DNA sequence of length l is defined as $GC(l)$ in (3), and $|\ G+C\ |$ represents the sum number of G and C. In this study, $|G+C\ |$ was assigned the value$\lfloor l/2 \rfloor$:

$$GC(l) = |G+C|/|l| \tag{3}$$

D. No-Runlength Constraint

The no-runlength constraint is used to avoid homopolymers in DNA storage codes. In DNA, the term homopolymer refers to the repetition of adjacent bases. Homopolymers increase the complexity of DNA synthesis, hence complicating assembly methods and increasing synthesis costs. Moreover, the repetition of bases and high GC-content will lead to the formation of secondary structure, preventing the elongation of DNA strands. The presence of homopolymers will increase the error rate of Illumina and nano sequencing [34] and also easily lead to the slip of polymerase. For example, *ATCCCG* may be misread as *ATCG* or *ATCCG* in sequencing, resulting in the loss of stored information.

The no-runlength constraint requires that there is a DNA sequence $L\ (l_1,\ l_2,\ l_3 \ldots, l_n)$ of length n such that (4) is satisfied for any i:

$$l_{\mathrm{i}} \neq l_{\mathrm{i}+1} \qquad \mathrm{i} \in [1, n-1], \tag{4}$$

E. Uncorrelated of the Address Constraint

In order to ensure accurate addressing and avoid costly post-processing, the address bits require a special encoding, which can be imposed by uncorrelated of the address constraint [21]. In DNA storage addressing, prefix/suffix matching may cause errors in information retrieval and sorting. The uncorrelated of the address constraint is inspired by mutually uncorrelated (MU) codes, which satisfy the constraint that the prefix and suffix sets of all code words do not intersect.

The prefix code is similar to the MU code, where prefix codes are a code system characterized by a "prefix property" that requires that no entire code word in the system is a prefix of any other code word in the system (initial segment). Mutually unrelated (MU) codes were originally studied by Levenshtein [35] for synchronization purposes and have recently gained attention due to their relevance and applicability to DNA storage. Any sequence in the coding set constructed by uncorrelated of the address constraint does not have a sufficiently long suffix as a prefix for another sequence [36], and vice versa. For a pair of DNA sequences, $A\ (a_1,\ a_2,\ a_3 \ldots\ a_n)$ and $B\ (b_1,\ b_2,\ b_3 \ldots\ b_n)$, the suffix of A cannot be used as the prefix of B, and vice versa. The prefix/suffix length is defined as s, the sequence $(a_1,\ a_2 \ldots a_s) \neq$ sequence $(b_{n-s+1}, b_{n-s+2} \ldots b_n)$ and the sequence $(b_1, b_2 \ldots b_s) \neq$ sequence $(a_{n-s+1}, a_{n-s+2} \ldots a_n)$. In this work, $s = 3$. For example, ATGCT and CGATG cannot appear at the same time because ATG is correlated.

III. Algorithm Description

A. Multi-Verse Optimizer (MVO) Algorithm

The MVO algorithm, a heuristic algorithm proposed by Mirjalili [37], is based on the multiverse theory and is used in many fields. In a multiverse, multiple universes attract or collide with each other. Under the guidance of the inflation rate, multiple universes eventually can reach a stable state through the exchange of matter through black/white holes and wormholes. The mechanism of the MVO algorithm is to increase the average fitness value by exchanging objects, and finally the system reaches stability. First, sort according to the standardized inflation rate, and then use roulette to determine which universe contains white holes. Formula (5), shown at the bottom of the page, is used to explore the search space through the black hole and white hole mechanisms. The specific expressions of wormhole existence probability (*WEP*) and travelling distance rate (*TDR*) are shown in (6) and (7), at the bottom of the page. With the iteration of the algorithm, the inflation rate of the universe increases, and the possibility of transferring objects increases. Besides, wormholes can appear randomly in all universes without considering the inflation rate. Under the premise of efficient exploration of the universe, it is assumed that the wormhole is built in the current universe and the universe with the best inflation rate. The relevant formulas are as follows, and the specific details are consistent with the original text.

B. Brownian Multi-Verse Optimizer (MVO) Algorithm

In the MVO algorithm, there are two ways to update the universe under the guidance of the expansion rate. In the process of using wormholes to transfer matter to update the universe, it mainly relies on the WEP of the current optimal universe. However, in the case where the initial position of the initial stage is not good, for example, the difference between the initial value and the optimal value is too large, and WEP cannot guide the population update iteration well. Brownian motion, which refers to the random movement of tiny particles or particles in a fluid, is introduced into the MVO algorithm to make up for the shortcomings of MVO. The Brownian motion [38] process is a normally distributed independent incremental continuous random process. It is one of the basic concepts in random analysis. Its basic properties

$$x_y = \begin{cases} \left[X_j + TDR \times \left(\left(ub_j - lb_j\right) \times r4 + lb_j\right), & r3 < 0.5 \quad r2 < WEP \\ X_j - TDR \times \left(\left(ub_j - lb_j\right) \times r4 + lb_j\right), & r\ 3 \geq 0.5 \\ x_y & r\ 2 \geq WEP \end{cases} \tag{5}$$

$$WEP = \min + (\max - \min) \times \frac{l}{L}, \tag{6}$$

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}}, \tag{7}$$

are as follows: Brownian motion $W(t)$ is a normal random variable with an expectation of 0 and a variance of t. For any $r \leq s$, $W(t) - W(s)$ is independent of $W(r)$ and is a normal random variable with an expectation of 0 and a variance of $t - s$. The governing probably density function at point x for this motion is expressed in (8). In the MVO algorithm, we treat the universe as tiny particles that undergo Brownian motion after being updated early in the iteration, and the results obtained go to the next iteration:

$$f_B(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{8}$$

The Nelder–Mead method, proposed by John Nelder and Roger Mead [39] in 1965, is a method for nonlinear optimization problems normally used to solve the minimization problem. The n-dimensional Nelder–Mead includes $n + 1$ test points. The Nelder–Mead algorithm forms a simplex, calculates the objective function value of each point to find a new better test point to replace the old one, and finally reaches the best point after the final iteration. The most commonly used method is to replace the worst point with the reflection point of the centroid (the average of the first n points). If the reflection point is better than the current point, the algorithm will continue to search in the direction of the reflection point. If the current point is better, then all points will shrink in a better direction. There are three strategies for replacing the worst value of the function: reflection, expansion and contraction. When the above three methods fail, shrink is used until the radius of the simplex is sufficiently small. There can be many definitions of radius here, such as the distance between two points, or the value of the largest dimension in a vector of two points. In this paper, when the Euclidean distance tends to 0, the simplex tends to a point, and this point is used as the output result.

The update strategy of the Nelder–Mead method and Brownian movement is applied in the BMVO algorithm instead of the random update based on the current global optimum, which can reduce the influence of individuals with local minimum or local maximum on the update mechanism. This strategy has the ability to accelerate convergence and continue the optimization process. Fig. 1 and Algorithm 1 show the details of BMVO and the entire process.

C. Experiment Environment

This work is conducted with Windows 10 on an i7 4790 8g RAM platform, using MATLAB 2018a for simulation experiments. The test function results of the algorithm are shown in Tables I–IV, and the results of the DNA coding set are shown in Tables VI–VII. The four bases were mapped to the four numbers 0–3 in the DNA coding set ($T \rightarrow 0$, $C \rightarrow 1$, $G \rightarrow 2$, $A \rightarrow 3$) in simulation experiments. The parameters used in this paper are the same as in the original paper.

D. Benchmark Functions

When designing heuristic algorithms, the benchmark function is a good method for verifying the performance of

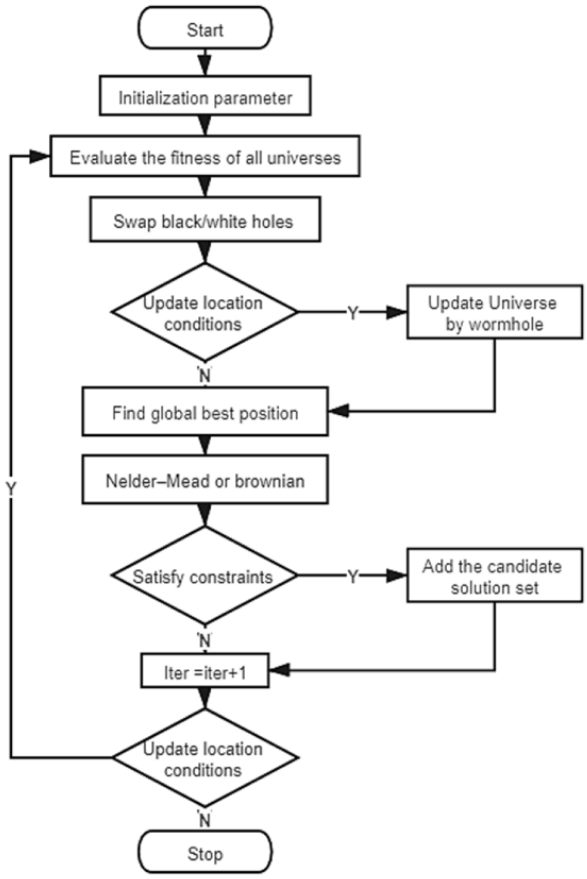

Fig. 1. BMVO algorithm flowchart.

the algorithm. The optimization performance of an algorithm can be reflected from the following aspects: convergence rate, precision, robustness and general performance. Because different algorithms may perform well in different conditions, a particular algorithm may not achieve ideal results for all problems. In this work, we selected 13 test functions [40-42], including seven high-dimensional unimodal functions and six high-dimensional multimodal functions, to evaluate algorithm performance. These benchmark functions can represent most of the optimization problems in practical applications and better reflect the performance of algorithm optimization and problems in the optimization process.

After obtaining the results of 30 function runs, the average and standard deviation are compared with the original algorithm and other representative algorithms. We chose to compare GA [43], PSO [44], GSA [45], GWO [46], MVO [37] and DMVO algorithms, where GA is the earliest and most enduring evolutionary algorithm, PSO is the beginning and most representative algorithm to imitate group behavior, GSA is based on inspired algorithms in physics, GWO is an algorithm with excellent performance in recent years and DMVO is the latest improvement based on MVO. The number of iterations is 500, and the dimension is 50. In order to ensure fairness, the results of MVO, GA, PSO, GSA and GWO data come from the results of Mirjalili [37]. Then, all algorithms pass Wilcoxon's non-parametric rank-sum detection to statis-

Algorithm 1 BMVO algorithm pseudo-code

```
For each universe indexed by i
    Update WEP and TDR
    Black_hole_index = i;
    For each object indexed by j
      r1 = random([0,1]);
      If r1 < NI(U i)
        White_hole_index = RouletteWheelSelection(-NI);
        U(Black_hole_index,j) = SU(White_hole_index,j);
      end if
      r2 = random([0,1]);
      If r2 < Wormhole_existence_probability
        r3 = random([0,1]);
        r4 = random([0,1]);
        If r3 < 0.5
          U(i,j) = Best_universe(j) + Traveling_distanc
          _rate * ((ub(j)- lb(j)) * r4 + lb(j));
        else
          U(i,j) = Best_universe(j)-Traveling_distance
          _rate * ((ub(j)- lb(j)) * r4 + lb(j));
        end if
      end if
    end for
    If Time < MaxTime/2
      Buniverse = Nelder–Mead (Best_universe);
    else
      Buniverse = brownian(dim) * Best_universe;
    end if
end for
```

tically evaluate the results. When $P > 0.05$, and the result of the algorithm is statistically significant.

The general test of the algorithm typically uses the high-dimensional unimodal functions (F1–F7), each of which has a global optimum. F8–F13 each has a global optimum and multiple local optima, and as the dimension increases, the number of local optimal solutions also increases. The specific function expression is shown in Table S1&S2 in the supplementary file. The high-dimensional multimodal function increases the difficulty of solving heuristic algorithms but can better reflect the optimization ability and robustness of an algorithm.

IV. Results Comparison and Analysis

A. Simulation Results of BMVO Algorithm

1) High-Dimensional Unimodal Function: In order to verify the performance of the BMVO algorithm, in this paper, 13 test functions are used for verification and evaluation. F1–F7 are high-dimensional unimodal functions. Their characteristics are that they are not too complicated, have a global optimum and are often used for general performance verification of algorithms. The number of iterations of all algorithms is set to 500. Tables I and II list the results after 30 runs. The results of MVO, GWO, GSA, PSO and GA all come from the paper of Mirjalili [37]. The simulation experimental results are shown in Table I and Table II, in which Table I and Table II respectively show the mean value and standard deviation of benchmark function results. It can be seen from the table that BMVO algorithm achieves ideal results in almost all functions. Among them, F1 and F3 have improved by 60 orders of magnitude compared with the original best results, which proves the good optimization ability of BMVO. Furthermore, the smaller standard deviation can reduce the accidental interference of the simulation results. This proves that the algorithm has high exploration ability, which is due to the combination of WEP and Brownian motion to assist BMVO in exploration. In order to make the results clearer, the convergence curves of F5 and F6 and the original algorithm are compared as shown in Fig. 2 and Fig. 3.

2) High-Dimensional Multimodal Function: F8–F13 are high-dimensional multimodal functions that have a global optimum and several local optima, which greatly increases the algorithm's ability to seek optimization compared with unimodal functions. Tables III and IV show the results after 30 runs, with data sources as described above. In the tables, we can clearly see that the BMVO algorithm has achieved the ideal result and found the global optimum in F9 and F11. In addition, through the average and standard deviation of F10 and F12, it can be seen that in the process of iteration, the BMVO algorithm jumps out of a local optimum and continues to iteratively converge, which illustrates the necessity of adding the Brownian motion and Nelder–Mead method to the BMVO algorithm. In F13, although the result is smaller than that of DMVO, it is still far greater than that of KMVO. The result of the algorithm is not ideal. This may be due to the special periodicity of the test function F13, which is also the direction that the next step can continue to improve.

3) Wilcoxon Rank-Sum Test: The rank-sum test is a non-parametric test, and it does not depend on the specific form of the overall distribution. It can be applied regardless of the distribution of the object that is being studied and has strong practicality. Sign test t can test whether there is a significant difference in paired test data in the case of an arbitrary overall distribution. But the sign test only considers the sign of the difference, not the absolute value of the difference. This strategy will result in the loss of part of the test information and a relatively rough result. In order to avoid this defect of the sign verification method, Wilcoxon proposed an improved method called the Wilcoxon rank-sum test [47]. This method considers both the direction of the difference and the size of the difference, which is more effective than the sign test. In this article, the inspection level is 0.05. That is, when $P > 0.05$, the result is considered to be statistically significant, and there is no difference in the distribution position of the group where the test data are located. In Table V, $P > 0.05$ is satisfied in most cases, which shows that the results are convincing and indicates the statistical significance of the BMVO algorithm.

B. Results of DNA Storage Coding Set

1) Comparison Coding Set Lower Bound: In a traditional electronic storage system, each storage block is divided into a boot area (address bit), a data area and other check bits. The DNA storage system is also divided into similar parts, and the common DNA single-strand storage structure is shown in Fig. 4. In addition to the payload, there are some non-payload

TABLE I
AVERAGE RESULTS OF UNIMODAL BENCHMARK FUNCTIONS

F	BMVO Avg	DMVO Avg	KMVO Avg	MVO[37] Avg	GWO[37] Avg	GSA[37] Avg	PSO[37] Avg	GA[37] Avg
F1	2.68E-85	1.67E-17	8.8094	2.08583	2319.19	2983.667	3.552364	27,187.58
F2	1.10E-46	6.83E-10	3.1771	15.92479	14.43166	10.96518	8.716272	68.6618
F3	6.73E-85	5.75E-17	2209.7671	453.2002	7278.133	113,740.40	2380.963	48,530.91
F4	3.06E-45	2.87E-09	1	3.123005	13.09729	32.2563	21.5169	62.99326
F5	48.4227	48.4392	603.1599	1272.13	3,425,462	7582.498	1132.486	65,361,620
F6	4.91E-03	0.17774	9.3446	2.29495	5009.442	74,617.45	86.62074	49,574.10
F7	2.68E-85	1.45E-04	0.11137	0.051991	0.408082	21.16092	0.577434	18.72524

TABLE II
SD RESULTS OF UNIMODAL BENCHMARK FUNCTIONS

F	BMVO SD	DMVO[37] SD	KMVO[37] SD	MVO[37] SD	GWO[37] SD	GSA[37] SD	PSO[37] SD	GA[37] SD
F1	1.47E-84	1.87E-17	1.6306	0.648651	1237.109	903.3827	2.853733	2745.82
F2	6.00E-46	3.80E-10	0.72651	44.7459	5.923015	10.54968	4.929157	6.062311
F3	2.69E-84	5.08E-17	786.9794	177.0973	2143.116	78,786.15	1183.351	8249.75
F4	1.28E-44	1.37E-09	0	1.582907	11.3469	6.226765	6.71628	2.535643
F5	3.44E-02	0.081404	611.1058	1479.477	3,304,309	7314.818	1357.967	29,714,021
F6	3.22E-03	0.080863	2.3629	0.630813	3028.875	8231.224	147.3067	8545.149
F7	1.47E-84	1.61E-04	0.028805	0.029606	0.119544	12.1566	0.318544	4.935256

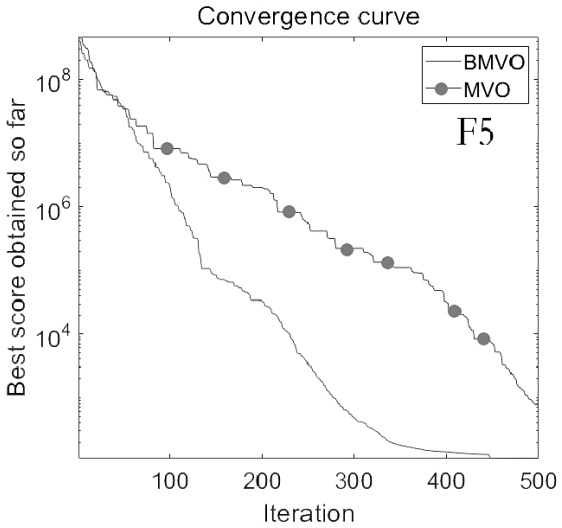

Fig. 2. The convergence curve is compared at F5.

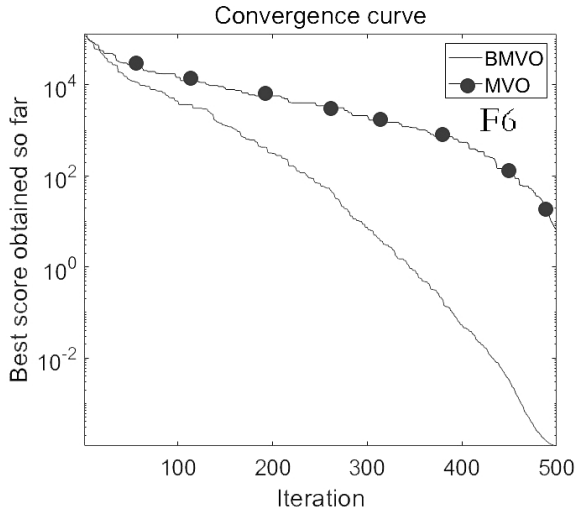

Fig. 3. The convergence curve is compared at F6.

bits, such as address bits, parity bits and primers. They are the concerns of this article, where the address bits are more important, because the correct addressing of the address bits is the key to complete data reading. However, in previous work [11], no attention was paid to the targeted coding of address blocks. In this work, the BMVO algorithm is used to design a DNA non-payload bit coding set that satisfies combination constraints and thermodynamic constraints.

$A^{GC,NL,UA}(n, d, w)$ is defined as a DNA storage coding set of length n and edit distance d, which satisfy the storage edit distance constraint, GC-content constraint, no-runlength constraint and uncorrelated of the address constraint. Table VI shows the lower limit of the number of codes when $4 \leq n \leq 10$ and $3 \leq d \leq n$. The superscript b is used to indicate the results from this project, and u indicates the best results from our previous study [48]. It can be clearly seen from Table VI that in most cases, the results in this article are significantly better. The results show that the lower limit of the code set constructed by the BMVO algorithm has been increased by 4%–16%. For example, when $n = 9$ and $d = 4$, it is increased by 16%.

TABLE III
AVERAGE RESULTS OF MULTI-MODAL BENCHMARK FUNCTIONS

F	BMVO	DMVO	KMVO	MVO[37]	GWO[37]	GSA[37]	PSO[37]
	Avg	Avg	Avg	Avg	Avg	Avg	Avg
F8	-13560.06	−12473.8746	−12,348.6192	−11,720.20	-10,739.50	-4638.41	-6727.59
F9	0	0	49.9577	118.046	89.13475	128.0103	99.83202
F10	8.88E-16	5.89E−10	2.9395	4.074904	9.452571	1.654073	4.295044
F11	0	0	0.75229	0.938733	22.51942	1021.705	624.3092
F12	6.07E-04	0.073642	1.898	2.459953	3,200,008	741,596.90	13.38384
F13	2.41E-04	4.68E+00	1.35E−32	0.222672	7,815,082	6,670,046	21.11298
F8	-13560.06	−12473.8746	−12,348.6192	−11,720.20	-10,739.50	-4638.41	-6727.59

TABLE IV
SD RESULTS OF MULTI-MODAL BENCHMARK FUNCTIONS

F	BMVO	DMVO	KMVO	MVO[37]	GWO[37]	GSA[37]	PSO[37]
	SD	SD	SD	SD	SD	SD	SD
F8	1.25E+03	693.532	724.6721	937.1975	1162.793	805.0488	1352.882
F9	0	0	0.017879	39.34364	37.95765	26.90054	24.62872
F10	0	2.53E-10	0.57575	5.501546	3.467608	1.583499	1.308386
F11	0	0	0.049873	0.059535	26.68168	82.95486	105.3874
F12	2.14E-04	0.057076	0.61601	0.791886	6,746,208	624,375.50	8.969122
F13	9.95E-05	5.38E-02	5.57E−48	0.086407	16,475,640	5,719,826	12.83179

TABLE V
P VALUES OF WILCOXON RANK-SUM TEST OVER 30 RUNS

F	BMVO	KMVO	MVO[37]	GWO[37]	GSA[37]
F1	N/A	0.34817	N/A	0.002827	0.000183
F2	0.49555	0.51033	0.009108	0.053903	0.909722
F3	0.50123	0.87769	N/A	0.000183	0.000183
F4	0.51126	N/A	N/A	0.140465	0.000183
F5	0.52866	N/A	0.677585	0.10411	0.005795
F6	0.50475	0.58492	N/A	0.000183	0.000182
F7	0.48473	0.62052	N/A	0.000183	0.000183
F8	0.5261	0.60117	N/A	0.053903	0.000183
F9	0.59996	0.26379	0.121225	N/A	0.002827
F10	N/A	0.35904	0.121225	0.001315	N/A
F11	N/A	0.59211	N/A	0.005795	0.000183
F12	0.44029	0.48449	N/A	0.025748	0.000183
F13	0.421	N/A	N/A	0.075662	0.000183

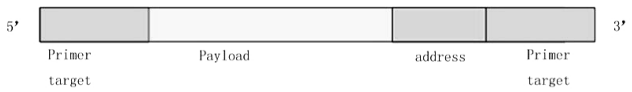

Fig. 4. Single stranded storage structure of DNA.

TABLE VI
LOWER BOUNDS FOR $A^{GC,NL,UA,MM}(n, d, w)$

n\d	3	4	5	6	7	8	9	10
4	6^u							
	$\mathbf{6}^b$							
5	12^u	5						
	$\mathbf{13}^b$	$\mathbf{5}^b$						
6	30^u	11^u	4^u					
	$\mathbf{31}^b$	$\mathbf{12}^b$	$\mathbf{4}^b$					
7	53^u	19^u	6^u	3^u				
	$\mathbf{56}^b$	$\mathbf{19}^b$	$\mathbf{6}^b$	$\mathbf{3}^b$				
8	101^u	38^u	13^u	4^u	3^u			
	$\mathbf{103}^b$	$\mathbf{41}^b$	12^b	$\mathbf{5}^b$	$\mathbf{3}^b$			
9	167^u	55^u	19^u	7^u	3^u	2^u		
	$\mathbf{173}^b$	$\mathbf{64}^b$	$\mathbf{19}^b$	$\mathbf{7}^b$	$\mathbf{3}^b$	$\mathbf{2}^b$		
10	250^u	110^u	34^u	11^u	5^u	3^u	2^u	
	$\mathbf{260}^b$	$\mathbf{120}^b$	$\mathbf{37}^b$	$\mathbf{12}^b$	$\mathbf{5}^b$	$\mathbf{3}^b$	$\mathbf{2}^b$	

$A^{GC,NL,UA,MM}(n, d, w)$ is defined as a DNA storage coding set that satisfies not only the above constraints but also the MFE constraint proposed in this work. Similarly, in Table VII, the lower bounds of the DNA storage coding set that satisfies the MFE constraint, storage edit distance constraint, GC-content constraint, No-runlength constraint and uncorrelated of the address constraint are also compared. It can be seen from Table VII that in most cases, the BMVO algorithm designs a larger lower bound. The superscript b indicates the results from this project, and u indicates the results from DMVO [48] under the same constraints. For example, in the

TABLE VII
LOWER BOUNDS FOR $A^{GC,NL,UA,MM}(n, d, w)$

n\d	3	4	5	6	7	8	9	10
4	2^u							
	4^b							
5	6^u	**3**^{u}						
	7^b	2^b						
6	17^u	6^u	2^u					
	16^b	**7**^{b}	**2**^{b}					
7	27^u	9^u	**4**^{u}					
	28^b	**11**^{b}	3^b					
8	47^u	**23**^{u}	6^u	2^u				
	48^b	21^b	**9**^{b}	**3**^{b}				
9	83^u	30^u	8^u	4^u	2^u			
	93^b	**33**^{b}	**12**^{b}	**5**^{b}	**2**^{b}			
10	122^u	57^u	19^u	6^u	2^u			
	137^b	**62**^{b}	**19**^{b}	**6**^{b}	**3**^{b}			

TABLE VIII
DNA STORAGE CODES SET IN $n = 10$, $d = 5$

TCTCTGCGTA	*CGTGTACATG*
TCTACGACAC	*CACTCACTGT*
TCACATCGCT	*CAGACTCACA*
TGCTGCTAGT	*GTGTGATCGA*
TGCGAGTCAT	*ATACTGTGCG*
TGCACACAGA	*ACGTAGTGCT*
TATGCGCTCA	*ACATGACGCA*
TAGCGTGATC	*ACACGCTATG*
TAGAGCACGT	*AGTCGTCAGT*
CTGTGTGTGT	

case of $n = 9$ and $d = 5$, the lower bound of BMVO is increased by 50% compared with the previous algorithm. However, in some cases, the improvement is not obvious, which may be due to stricter coding constraints causing the narrowing of the candidate solution set. Therefore, although a better algorithm is used, only a small part of the new codes can be obtained. However, under relatively loose constraints, the BMVO algorithm can still find more effective new codes. For example, when $n = 9$ and $d = 3$, the lower bound of BMVO increases by 10%, providing an effective improvement. In Table VIII, the DNA storage encoding set satisfying the combined constraint is given in the case of $n = 10$ and $d = 5$. And in the supplementary Document 1, all DNA storage code words constructed by the BMVO algorithm under two sets of constraints are given.

2) Comparison of Thermodynamic Properties: In addition, in order to demonstrate the encoding performance under MFE constraints, a thermodynamic comparison is made with other work. The simulation experiment is conducted under the condition that the DNA molecule concentration is 1 nM and the salt concentration is 1 M. Fig. 5 shows the comparison of TM variance of the DNA storage coding set constructed

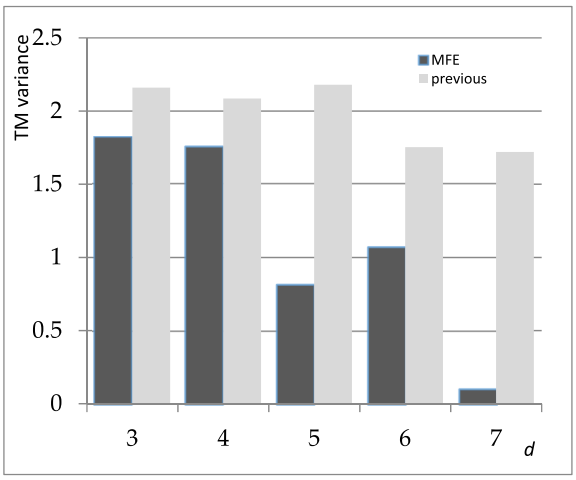

Fig. 5. TM under two conditions when $n = 9$.

TABLE IX
THERMODYNAMIC TM VARIANCE COMPARISONS

n/d		3	4	5	6	7
8	Previous[48]	2.3773	2.3382	1.7156	2.2094	0.5924
	MFE	**1.3675**	**1.7349**	**1.7944**	**2.7299**	NULL
9	Previous [48]	2.1592	2.0851	2.1794	1.7534	1.7198
	MFE	**1.8167**	**1.7498**	**0.8121**	**1.0674**	**0.1042**
10	Previous[48]	2.2507	1.9431	2.1939	**1.5834**	2.4125
	MFE	**1.4417**	**1.4703**	**1.6557**	1.6306	**2.1297**

with the BMVO before and after the MFE constraint is added, and length $n = 9$. The abscissa represents the distance d ($2 < d < 8$), and the ordinate represents the value of TM variance. The specific values of n and d in other cases are shown in Table IX. For easy control of DNA sequences and accuracy of storage results, the TM of all participating DNAs should be controlled within a small fluctuation range, i.e., the variance of TM should be minimized. The smaller the variance of TM will reduce the errors during the melting process. In the vast majority of cases, the TM variance of the coding set constructed by the combination constraints with the addition of MFE constraints is reduced by more than 7%, which is significant.

In Table X, the results with the address bit lengths of 6, 8 and 16 are compared among our work and the works of Carmean [49], Meiser *et al.* [11], Limbachiya *et al.* [50] and Yazdi *et al.* [51]. It can be seen from the table that when *address length* = 6, *MFEMAX*, *MFEMIN* and *MFEAVG* are significantly reduced, confirming the superiority of the coding method in this paper. When *address length* = 8, ideal results are obtained on the relevant attributes of MFE and TM. However, it can be seen that the code set constructed by Limbachiya [50] is larger. This is because the constraints in this paper are stricter than in the paper by Limbachiya, such as the addition of the uncorrelated of the address constraint and MFE constraint. When *address length* = 16, not only are the results of this paper better in terms of MFE, but also the TM variance is reduced by an order of magnitude, and the DNA storage coding set obtained in this project is also larger.

TABLE X
THERMODYNAMIC MFE AND TM COMPARISONS

	Address length	Address size	MFEMAX	MFEMIN	MFEAVG	TM variance
Carmean [49]	6	1	-5.3	-5.3	-5.3	NULL
Meiser [11]	6	4	-4.2	-5.5	-4.775	NULL
Our work	6	15	**-5.3**	**-6.5**	**-5.753**	NULL
Limbachiya [50]	8	289	-6.6	-9.5	-8.117	2.3733
Our work	8	47	**-8**	**-9.5**	**-8.4723**	**1.3675**
Yazdi [51]	16	17	-15	-23.9	-20.3	10.9729
Our work	16	113	**-19.7**	**-21.7**	**-20.3**	**1.1931**

The results in the table show that the coding set in this paper has achieved ideal results both in quality and quantity. In addition, the imbalance between homopolymer and GC-content often occurs in Yazdi's [51] address codewords, which may lead to an increase in error rate. In this work, this problem is solved by the GC-content constraint and no-runlength constraint.

It can be seen from Tables VI and VII that the BMVO algorithm increases the lower bounds of DNA storage codes by 4%–16% and 4%–50% under the different constraints, which provides a theoretical basis for the realization of DNA storage codes. The larger code set illustrates the superiority of the BMVO algorithm in practical applications. More choices of address bits mean more effective addressing can be provided, that is, more effective information is stored under the same physical capacity. At the same time, the TM variance of the DNA storage coding set resulting after adding the MFE constraint is reduced by more than 7%. A more stable TM means a smaller error rate in synthesis and sequencing. Finally, under the coding conditions of the same length, a thermodynamic comparison with the previous work shows the advantages of the coding method in practical application in this paper. A smaller MFE will make the reaction more stable and efficient. In other respects, computational models similar to neurons, such as spiking neural networks [52], next-generation future computing and parallel computing models [53], are considered to be used to design DNA coding sets. And the recently proposed theory of biological information security [54] and combining machine learning techniques with bioinformatics [55] is also worthy of in-depth study.

V. CONCLUSION

DNA melting and synthesis processes are all biochemical reactions. The essence of biochemical reactions is the change of energy, so MFE is considered in this article. The size of MFE can reflect the stability of DNA single strands to a certain extent and the error rate in the progresses of PCR and sequencing. Therefore, a new MFE constraint is proposed to be used in combination with the previous constraints to construct the DNA storage coding set by the BMVO algorithm. In the result analysis, the lower bound of the storage coding set, which is designed by the BMVO algorithm, is compared with the previous results. Before and after adding MFE constraints, the lower bounds of DNA storage coding increase by 4%–16% and 4%–50%, respectively. The results prove the potential of the BMVO algorithm in practical application. In order to prove the applicability of MFE constraints in DNA storage coding, *MFEMIN*, *MFEMAX* and *MFEAVG* and the variances of TM are compared through thermodynamic comparison experiments. MFE and TM variance are significantly reduced in most cases, especially when *address length* $= 8$, in which case *MFEMAX* and TM variance are reduced by 21% and 76%.

In this paper, based on Brownian motion, Nelder–Mead method and MVO algorithm, a BMVO algorithm is proposed to design DNA storage coding sets. The reason for using heuristic algorithms to design coding sets is that the amount of DNA data is large, the capacity of the candidate solution sets is large and the optimal solution cannot be obtained in polynomial time. Therefore, the DNA storage coding problem is abstracted as a multi-objective optimization problem, and a heuristic algorithm is used to find an approximate optimal solution. The BMVO algorithm uses Brownian motion and the Nelder–Mead method to improve slow iteration and single update method in the MVO algorithm. The quantitative comparison with mainstream heuristic algorithms (DMVO, GA, GWO, GSA and PSO) and the results of the test functions illustrate the superior performance of the BMVO algorithm. These results are shown in detail in Tables I–IV.

In future work, we will continue to focus on coding issues in DNA storage. The size and quality of the encoding set seem to be two opposing properties, but how to achieve a reasonable balance is a question worth considering. Compared with combination constraints, thermodynamic constraints can better reflect energy changes and reaction processes. In the construction of DNA storage coding, there are some cases in which candidate solutions with different constraints overlap, such as GC-content constraints and TM values, so we plan to use these constraints selectively. This plan not only reduces the computational complexity but also makes it possible to find a better balance between quality and quantity. In addition, we will continue to improve the algorithm to achieve better results, and we hope to apply the algorithm in other fields, such as artificial neural networks and engineering problems.

REFERENCES

[1] H. N. Poinar *et al.*, "Genetic analyses from ancient DNA," *Annu. Rev. Genet.*, vol. 38, no. 1, pp. 645–679, 2004.
[2] M. S. Neiman, "Some fundamental issues of microminiaturization," *Radiotekhnika*, vol. 1, no. 1, pp. 3–12, 1964.

[3] J. Davis, "Microvenus," *Art J.*, vol. 55, no. 1, pp. 70–74, 1996.
[4] C. Bancroft, T. Bowler, B. Bloom, and C. T. Clelland, "Long-term storage of information in DNA," *Science*, vol. 293, no. 5536, pp. 1763–1765, 2001.
[5] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "Toward a DNA-based archival storage system," *IEEE Micro*, vol. 37, no. 3, pp. 98–104, May/Jun. 2017.
[6] H. Nguyen *et al.*, "Long-term stability and integrity of plasmid-based DNA data storage," *Polymers*, vol. 10, no. 1, p. 28, Jan. 2018.
[7] K. J. Tomek *et al.*, "Driving the scalability of DNA-based information storage systems," *ACS Synth. Biol.*, vol. 8, no. 6, pp. 1241–1248, Jun. 2019.
[8] W. D. Chen *et al.*, "Combining data longevity with high storage capacity–layer-by-layer DNA encapsulated in magnetic nanoparticles," *Adv. Funct. Mater.*, vol. 29, no. 28, Jul. 2019, Art. no. 1901672.
[9] Y. Zhang *et al.*, "Encoding carbon nanotubes with tubular nucleic acids for information storage," *J. Amer. Chem. Soc.*, vol. 141, no. 44, pp. 17861–17866, Nov. 2019.
[10] Y. Wang, M. Keith, A. Leyme, S. Bergelson, and M. Feschenko, "Monitoring long-term DNA storage via absolute copy number quantification by ddPCR," *Anal. Biochem.*, vol. 59, pp. 8476–8480, Oct. 2019.
[11] L. C. Meiser *et al.*, "Reading and writing digital data in DNA," *Nature Protocols*, vol. 15, no. 1, pp. 86–101, 2020.
[12] R. N. Grass, R. Heckel, C. Dessimoz, and W. J. Stark, "Genomic encryption of digital data stored in synthetic DNA," *Angew. Chem. Int. Ed.*, vol. 59, no. 22, pp. 8476–8480, May 2020.
[13] M. H. Garzon and R. J. Deaton, "Codeword design and information encoding in DNA ensembles," *Natural Comput.*, vol. 3, no. 3, pp. 253–292, Aug. 2004.
[14] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, 2012.
[15] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.
[16] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angew. Chem. Int. Ed.*, vol. 54, no. 8, pp. 2552–2555, 2015.
[17] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," *ACM SIGOPS Operating Syst. Rev.*, vol. 50, no. 2, pp. 637–649, Mar. 2016.
[18] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–953, Mar. 2017.
[19] W. Song, K. Cai, M. Zhang, and C. Yuen, "Codes with run-length and GC-content constraints for DNA-based data storage," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 2004–2007, Oct. 2018.
[20] K. A. Schouhamer Immink and K. Cai, "Design of capacity-approaching constrained codes for DNA-based storage systems," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 224–227, Feb. 2018.
[21] S. M. H. Tabatabaei Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, "Mutually uncorrelated primers for DNA-based data storage," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6283–6296, Sep. 2018.
[22] L. Organick *et al.*, "Random access in large-scale DNA data storage," *Nature Biotechnol.*, vol. 36, no. 3, pp. 242–248, 2018.
[23] L. Song and A.-P. Zeng, "Orthogonal information encoding in living cells with high error-tolerance, safety, and fidelity," *ACS Synth. Biol.*, vol. 7, no. 3, pp. 866–874, Mar. 2018.
[24] Y. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, and C. L. Poh, "Construction of bio-constrained code for DNA data storage," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 963–966, Jun. 2019.
[25] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Commun.*, vol. 10, no. 1, Dec. 2019.
[26] P. Fei and Z. Wang, "LDPC codes for portable DNA storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 76–80.
[27] R. Appuswamy, "OligoArchive: Using DNA in the DBMS storage hierarchy," in *Proc. CIDR*, Jan. 2019, pp. 1–7.
[28] Q. Zhang, B. Wang, X. Wei, X. Fang, and C. Zhou, "DNA word set design based on minimum free energy," *IEEE Trans. Nanobiosci.*, vol. 9, no. 4, pp. 273–277, Dec. 2010.
[29] D. Tulpan *et al.*, "Thermodynamically based DNA strand design," *Nucleic Acids Res.*, vol. 33, no. 15, pp. 4951–4964, Sep. 2005.
[30] M. Andronescu, Z. C. Zhang, and A. Condon, "Secondary structure prediction of interacting RNA molecules," *J. Mol. Biol.*, vol. 345, no. 5, pp. 987–1001, Feb. 2005.
[31] B. Wang *et al.*, "Constructing DNA barcode sets based on particle swarm optimization," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 3, pp. 999–1002, May 2018.
[32] B. Wang, Q. Zhang, and X. Wei, "Tabu variable neighborhood search for designing DNA barcodes," *IEEE Trans. Nanobiosci.*, vol. 19, no. 1, pp. 127–131, Jan. 2020.
[33] X. Li, B. Wang, H. Lv, Q. Yin, Q. Zhang, and X. Wei, "Constraining DNA sequences with a triplet-bases unpaired," *IEEE Trans. Nanobiosci.*, vol. 19, no. 2, pp. 299–307, Apr. 2020.
[34] D. Laehnemann, A. Borkhardt, and A. C. Mchardy, "Denoising DNA deep sequencing data—High-throughput sequencing errors and their correction," *Briefings Bioinf.*, vol. 17, no. 1, pp. 154–179, 2016.
[35] V. I. Levenshtein, "Decoding automata which are invariant with respect to their initial state," *Problems Cybern.*, vol. 12, pp. 125–136, 1964.
[36] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for DNA storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3671–3691, Jun. 2019.
[37] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
[38] J. Topping, "Investigations on the theory of the Brownian movement," *Phys. Bull.*, vol. 7, no. 10, p. 281, 1956.
[39] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.
[40] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
[41] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, Jan. 2001.
[42] X.-S. Yang, "Test problems in optimization," 2010, *arXiv:1008.0549*. [Online]. Available: http://arxiv.org/abs/1008.0549
[43] R. Deaton, "Genetic search of reliable encodings for DNA-based computation," in *Proc. 1st Annu. Conf. Genetic Program.*, vol. 419, 1996, pp. 1–7.
[44] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
[45] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[46] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[47] D. H. Kim and Y. C. Kim, "Wilcoxon signed rank test using ranked-set sample," *Korean J. Comput. Appl. Math.*, vol. 3, no. 2, pp. 235–243, Jun. 1996.
[48] B. Cao, X. Li, X. Zhang, B. Wang, Q. Zhang, and X. Wei, "Designing uncorrelated address constrain for DNA storage by DMVO algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Jul. 27, 2020, doi: 10.1109/TCBB.2020.3011582.
[49] D. Carmean, L. Ceze, G. Seelig, K. Stewart, K. Strauss, and M. Willsey, "DNA data storage and hybrid molecular–electronic computing," *Proc. IEEE*, vol. 107, no. 1, pp. 63–72, Jan. 2019.
[50] D. Limbachiya, M. K. Gupta, and V. Aggarwal, "Family of constrained codes for archival DNA data storage," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 1972–1975, Oct. 2018.
[51] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 1, p. 5011, Jul. 2017.
[52] T. Song, X. Zeng, P. Zheng, M. Jiang, and A. Rodriguez-Paton, "A parallel workflow pattern modeling using spiking neural p systems with colored spikes," *IEEE Trans. Nanobiosci.*, vol. 17, no. 4, pp. 474–484, Oct. 2018.
[53] T. Song, L. Pan, T. Wu, P. Zheng, M. L. D. Wong, and A. Rodriguez-Paton, "Spiking neural p systems with learning functions," *IEEE Trans. Nanobiosci.*, vol. 18, no. 2, pp. 176–190, Apr. 2019.
[54] X. Zhang, Q. Zhang, Y. Liu, B. Wang, and S. Zhou, "A molecular device: A DNA molecular lock driven by the nicking enzymes," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 2107–2116, 2020.
[55] Q. Zou and Q. Liu, "Advanced machine learning techniques for bioinformatics," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 4, pp. 1182–1183, Jul. 2019.