Designing Uncorrelated Address Constrain for DNA Storage by DMVO Algorithm

Ben Cao, Xue Li, Xiaokang Zhang, Bin Wang, Qiang Zhang and Xiaopeng Wei

Abstract—At present, huge amounts of data are being produced every second, a situation that will gradually overwhelm current storage technology. DNA is a storage medium that features high storage density and long-term stability and is now considered to be a feasible storage solution. Errors are easily made during the sequencing and synthesis of DNA, however. In order to reduce the error rate, novel uncorrelated address constrain are reported, and a Damping Multi-Verse Optimizer (DMVO) algorithm is proposed to construct a set of DNA coding, which is used as the non-payload. The DMVO algorithm exchanges objects through black/white holes in order to achieve a stable state and adds damping factors as disturbances. Compared with previous work, the coding set obtained by the DMVO algorithm is larger in size and of higher quality. The results of this study reveal that the size of the DNA storage coding set obtained by the DMVO algorithm increased by 4–16%, and the variance of the melting temperature decreased by 3–18%.

Index Terms: DNA storage, DNA coding, DMVO, Nanopore sequencing

1 Introduction

Following the digital revolution, the world entered the information age. With the popularization of computer and electronic data, the era of big data has arrived. According to the "Data Age 2025" report released by the research firm IDC, the annual data generated worldwide has increased from 33 ZB in 2018 to 175 ZB, which is equivalent to 491 EB per day. Massive information requires researchers to consider ways to more effectively store information and use information efficiently [1]. These requirements necessitate a new medium with more efficient storage and higher storage capacity. As a high-density storage medium, DNA has long-term stability and is thus seen as a feasible solution. DNA consists of 4 nucleotides—adenine (*A*), thymine (*T*), cytosine (*C*), and guanine (*G*)—and its theoretical storage capacity is therefore twice that of the binary code used in traditional electronic systems. Studies have shown that DNA can be stored for 10,000 years under appropriate conditions [2]. The cost of DNA synthesis and sequencing was once the main impediment hindering DNA storage, but with the recent reduction in the cost of synthetic technology and the rapid development of nanopore sequencing technology, the use of DNA as a medium to store information has been initially realized in the laboratory [3]. Neiman [4] first proposed the concept of DNA-based data storage in the 1960s.

Currently, DNA-based data storage primarily utilizes 2 methods to store encoded DNA sequences: *in vivo* and *in vitro*. *In vivo* DNA-based data storage is often used for groundbreaking work. For example, Joe Davis [5] proposed an avant-garde "Microvenus" project using bacteria as a storage medium for non-biological information in 1988. At the beginning of this century, a simple coding method using codon triplets demonstrated the great potential of DNA as a storage medium [6]. Arita et al. [7] introduced a simple and practical method for embedding short tags into genomic DNA, and the first *Bacillus subtilis* was cultivated using engineering design standards. A coding table for the long-term storage and retrieval of data was proposed by Wong et al. [8], encoding information into artificial DNA strands and inserting them into live hosts. Nguyen et al. [9] used a Perl script to encode the DNA sequence of a 2,046-word document. After synthesizing the DNA sequence, the information was stored and integrated into a plasmid vector. However, the ability of bacteria to carry plasmids is limited by the type and size of the plasmid. In addition, plasmids in bacteria or other live hosts can be easily mutated.

In recent studies, DNA-based *in vitro* data storage has been more common than *in vivo* data storage. A DNA storage scheme that encodes messages in multiple ways was reported by Church et al. [10] in order to avoid difficult-to-read sequences such as extreme *GC* content, repetitions, or secondary structures. Goldman et al. [11] described a scalable method that improves upon previous DNA-based information storage methods, which encoded only a small amount of information. This technique can reliably store more information than previous approaches and was able to reconstruct the original document with 100% accuracy. Chen [12] used silica pellets as DNA storage containers and increased the actual density of long-term DNA storage by designing alternating DNA layers

- *B. Cao, X. Li ,X. K. Zhang, B. Wang and Q. Zhang was with Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Ministry of Education, School of Software Engineering, Dalian University, Dalian, 116622, China (e-mail: carebon@163.com , daisywowa@gmail.com , xiaokangz96@gmail.com , wangbinpaper@gmail.com , zhangq@dlut.edu.cn).*
- *Q. Zhang and X.P. Wei are with School of Computer Science and Technology, Dalian University of Technology, 116024, Dalian, China (e-mail: zhangq@dlut.edu.cn, xpwei@dlu.edu.cn).*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

and polycationic layers in a layer-by-layer (LbL) structure. In order to quantitatively analyze the process of DNA storage, state-of-the-art droplet digital PCR (ddPCR) technology can be used to monitor the long-term storage of DNA [13]. A ddPCR reaction is split into approximately 20,000 nanometer-sized droplets. Each droplet is then analyzed for an absolute DNA copy number that can be quantified. Heckel et al. [14] hoped to break through technical limitations by providing quantitative and qualitative analyses to help guide the design of future DNA data storage systems. Carmean et al. [15] proposed a hybrid molecular-electronic architecture to take advantage of these 2 areas of molecular processing and storage. Meiser et al. [16] used error correction codes to protect information, thereby enabling error-free storage. In their work, they provided technical details and precise instructions for converting digital information into DNA sequences, physically processing biomolecules, storing biomolecules, and then regaining information through DNA sequencing. *In vitro* DNA-based data storage, however, requires a larger amount of DNA oligonucleotides than *in vivo* DNA data storage [17].

The most important and difficult aspect of DNA storage is the synthesis and sequencing of DNA strands. Of course, with the development of synthetic sequencing technology, both the expense and the technical difficulties have been reduced, although the problem of low sequencing accuracy still exists [18, 19]. Nanopore sequencing is the fourth generation of sequencing technology and uses single-molecule DNA (RNA) to infer base composition through nanopore current changes [20]. Since nanopore sequencing technology does not require the chain reaction of DNA polymerase, the problem of DNA polymerase inactivation does not exist. In theory, as long as the DNA sequence is stable enough, it can always be sequenced through the nanopore, and the sequencing length is greatly increased. Using longer sequences for DNA storage can reduce costs and data redundancy. At present, the application length of nanopore sequencing in the human genome and *E. coli* has reached 1 MB [21]. In the nanopore sequencing process, however, some problems still remain to be solved. Nanopore sequencing not only requires a sufficiently stable long sequence, but it is also prone to substitution, deletion, and other errors in the sequencing process. The accuracy rate of 1D nanopore sequencing is about 86% [22–23]. However, when sequencing both positive and negative strands of DNA while simultaneously ensuring the coding quality of both the DNA strand and the complementary strand in nanopore sequencing, i.e., the $1D^2$ sequencing method, the accuracy rate can reach 96%.

In the field of traditional electronic storage, data are encoded as either 0 or 1 and stored to disks via electromagnetic signals. In terms of DNA storage, coding is also the first process to consider. Effectively encoding DNA sequences can not only improve throughput and storage capacity but also reduce the chances of errors during storage. Reasonable encoding is irreplaceable in terms of the data integrity and robustness of the DNA storage system. For these reasons, our research focuses on methods for constructing a reasonable and efficient DNA storage coding set.

The problem of the combination constraint in DNA coding was first proposed by Garzon et al. [24]. The primary challenge in the coding of biological information is to design an effective and reasonable DNA sequence library to prevent non-specific hybridization during the reaction process. These researchers also discussed the method of constructing a DNA sequence library with thermodynamic and combination constraints and presented both experimental and theoretical results. In the error correction process, error-correcting coding in communications has been applied to DNA-based data storage [25]. The need for error correction coding was emphasized, and 83 KB of text data was encoded into DNA using silicon material, resulting in improved coding efficiency. Blawat et al. [26] proposed a coding scheme for errors generated during DNA sequencing, amplification, and synthesis (such as insertion, deletion, and substitution). The successful storage and retrieval of 22 MB of error-free data in the experiment proved the feasibility of DNA as a storage medium. This error correction scheme, however, did not detect and correct single-base errors. Gabry et al. [27] introduced asymmetric Lee distance (ALD) coding to correct errors in DNA-based storage systems and systems with parallel string transfer protocols, demonstrating effective construction and near-optimal redundancy. In addition, there are coding methods for converting binary sequences into DNA base sequences (codeword) [28]. Under the condition that the symbol sequences of A, T, C, and G meet 2 constraints, a 1.9 bit/base rate has been realized under the condition of low encoding/decoding complexity and limited error propagation. Immink et al. [29] described a simple coding scheme that avoided the appearance of long homopolymers and constructed a coding rate close to the theoretical maximum. Their proposed k-constrained q-ary data sequence replacement method features significant improvements in coding redundancy over existing k-constrained binary data sequence replacement methods. Yazdi et al. [30] proposed a weakly mutually uncorrelated (WMU) sequence for primer design. The desirable characteristic of a WMU sequence is that one sequence does not have a sufficiently long suffix to be the prefix of the sequence itself or another sequence. WMU sequences also require a large mutual Hamming distance from each other, have a balanced symbolic composition, and avoid primer-dimer byproducts. Wang et al. [31] proposed a new content-balanced run-length limit code (C-RLL), which can simultaneously generate short DNA sequences that satisfy 2 constraints. They have also developed a coding method with a high effective bit rate and low coding complexity that maps binary data into DNA sequences for DNA data storage. This technique not only meets biochemical constraints but also guarantees the stability of DNA sequences. Song et al. [32] proposed a novel self-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

error-detecting, 3-base block encoding scheme (SED3B) for reliable live cell and orthogonal information coding. SED3B is an innovative method of adding error detection bases into small data blocks that utilizes the inherent redundancy of DNA molecules for effective error correction. In this research, 10 different bar codes were encoded in *E. coli*. After 10 days of exposure to ultraviolet light, all of the bar codes were fully recovered, thus proving the stability of the encoded information. An enzyme synthesis strategy for data storage developed by Lee [33] employs template-independent polymerase terminal deoxynucleotidyl transferase (TdT). A digital codec was further designed to reduce the requirements of synthesis accuracy and sequencing coverage, providing both a method and a theoretical basis for the development of DNA digital information storage technology. In 2020, Yin [34] et al. proposed NOL-HHO algorithm to construct DNA storage coding set, and they maintain a smooth transition between exploration and exploitation through NOL strategy. On the problem of DNA coding, the construction of a larger coding set promotes the further development of DNA storage technology.

In the nanopore sequencing of DNA storage, address bit error is fatal compared to that of the data bit and will cause the entire string of data to be lost or misread. In traditional methods, address information is added to each individual sequence using random address encoding. For example, this information is randomly generated using a pseudo-random key generated by seeds [35]. Goldman et al. [11] used code tables to generate address coding in their research, although this coding method can only ensure that homopolymers do not appear in the address coding and requires further processing. In the current study, a new constraint for constructing non-payload coding sets from the coding point is proposed in order to reduce the probability of errors in nanopore sequencing. At the same time, a new storage edit distance constraint is introduced that focuses on the total number of add, remove, and replace operations, which is similar to the nanopore sequencing error. This, along with the new uncorrelated address constraint, the *GC* content constraint, and the No-runlength constraint, constitutes the combinatorial constraint, which ensures the quality of the DNA storage coding set. The Damping Multi-Verse Optimizer (DMVO) algorithm constructs the DNA storage coding set that satisfies the combination constraint. The DMVO algorithm improves the Multi-Verse Optimizer (MVO) algorithm by employing the damping factor and Lévy flight. Specifically, this method improves the diversity of the population via the damping factor and reduces the dependence on the current optimal individual during updating via Lévy flight.

The organization of this article is as follows. Section 2 describes the constraints of the coding set in DNA storage and delineates the individual constraints. The mechanism of the MVO algorithm is introduced in Section 3, as well as improvements based on Lévy flight and the damping factor. Lévy flight is a kind of random walk with a heavy-tailed probability distribution. The damping factor is a measure describing how rapidly oscillations decay from one bounce to the next. Section 4 includes the results and analysis of the test function, along with the analysis and comparison of the final DNA coding set. Section 5 summarizes this study and presents an overall outlook.

2 Constraints in DNA Storage Coding

The construction of a set of DNA strands of a given length n in DNA storage is designed not only to make more efficient use of DNA as a storage medium but also to avoid errors in synthesis and sequencing. In this study, the coding of non-payload sets in DNA storage was mainly performed in order to avoid errors in data reading.

2.1 Hamming distance constraint

For any pair of different DNA sequences α , β in the set, the Hamming distance constraint [36] is denoted as $\mathrm{H}(\alpha,\beta)\geq \mathrm{d}$, where $\mathrm{H}(\alpha,\beta)$ denotes the number of elements different from each other between α and β [28]. The Hamming distance is calculated as

$$\mathrm{H}(\alpha,\beta)=\sum_{i=1}^{n} h(\alpha_i,\beta_i), h(\alpha_i,\beta_i)=\begin{cases}0, \alpha_i=\beta_i\\ 1, \alpha_i\neq\beta_i\end{cases} . \quad (1)$$

The Hamming distance is used to describe the magnitude of the similarity of the 2 sequences; the higher the value, the lower the similarity. This means that the fewer different bases there are between 2 DNA sequences, the more of the same base there is. Therefore, there is a greater likelihood of non-specific hybridization between the DNA sequences.

2.2 Storage edit distance constraint

In information theory, edit distance is often used to measure the difference between 2 sequences. Succinctly, the edit distance between 2 strings is the sum of the minimum number of single-character edits (add, delete, and replace) [37]. Editing distance can also be used in natural language processing and bioinformatics to judge the similarity of DNA composed of *ACGT*. In the alphabet $\sum=\{\mathrm{A,C,G,T}\}$, there is a set U of length n and size $|\mathrm{U}|=4^n$. A subset $\mathrm{V}\subseteq\mathrm{U}$, 2 strings in V, and u, v satisfy

$$\eta(u,v)\geq d , \quad (2)$$

where d is a positive integer and η is the constraint of the DNA storage coding set, where the constraint is the edit distance.

In order to properly handle the problem of addition, deletion, and replacement errors in nanopore sequencing, the edit distance is introduced into the constraints of DNA storage coding. When applied to the DNA section that is to be sequenced using nanopore sequencing, the possibility of errors can be reduced. Even if an error occurs at a certain base, an address error can be avoided, since there may be no matched sequence in the DNA address coding set.

The storage edit distance is defined as follows for DNA codewords u, v of length n. $\mathrm{G}(u, v)$ defines the editing distance between u and v, and $\mathrm{UE}(u_i)$ defines the stor-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

age editing distance. UE(u_i) defines the minimum G(u_i, v_j) in all DNA coding sets, which should not be greater than the element d:

$$\mathrm{UE}(u_i)= \min_{1\le j\le n,\, j\neq i} \{G(u_i, v_j)\} \ge d \tag{3}$$

2.3 *GC* content constraint

The DNA sequence is composed of 4 bases, and the *GC* content constraint defines the ratio of the sum of *G* and *C* content to the total number of bases in the entire chain [38]. Generally, sequences with a *GC* content of approximately 50% are relatively stable. The *GC* content of length *s* is defined as *GC(s)*, and the calculation method is as follows, where | G+C | represents the sum of the numbers of *G* and *C*, respectively. In this study, the |G+C| sum is set to $\lfloor s/2 \rfloor$.

$$GC(s) = |G + C|/|s| \tag{4}$$

2.4 No-runlength constraint

Coding in DNA storage should not contain duplicate bases. Whether during synthesis or sequencing, repeated nucleotides can cause errors [39, 40]. For example, in *AGCCCT*, *C* is repeated and is easily misread as *AGCT* or *AGCCT* during sequencing, resulting in the loss of DNA storage information and thus generating reading and writing errors. This constraint requires that there is a DNA sequence S (s_1, s_2, s_3. .., s_n) of length n such that for any i,

$$S_i \neq S_{i+1} \qquad i \in [1, n-1] \tag{5}$$

2.5 Uncorrelated address constraint

An address is used to provide a unique index for a data block, so address substrings should not be similar. In addition, prefix/suffix matching may lead to errors in reading data in the block during information retrieval and sorting [30][41]. For 2 DNA sequences L (l_1, l_2, l_3. .. l_n) and H (h_1, h_2, h_3... h_n), the suffix of L cannot appear as the prefix of H, and vice versa. The prefix/suffix length is defined as s, i.e., the sequence (l_1, l_2. .. l_s) ≠ sequence (h_{n-s+1}, h_{n-s+2}. .. h_n) and the sequence (h_1, h_2... h_s) ≠ sequence (l_{n-s+1}, l_{n-s+2}. .. l_n). In this study, the prefix/suffix length is defined as 3 (s = 3). For example, *ATGCT* and *CGATG* cannot appear at the same time at which *ATG* is correlated.

During the process of sequencing and assembly, there is a DNA address coding set $A(a_1,a_2,a_3,a_4)$ consisting of (*ACGCATG*, *TCATACG*, *CTATGTC*, *GATACGC*), where a_1 and a_2 may lead to read assembly errors in blocks during joint informational retrieval and sequencing. The coding set *B* (*ACGCATG*, *CTATGTC*, *GATACGC*) is obtained after coding set *A* passes the uncorrelated address constraint. Coding set *B* can eliminate address cross-hybridization and cross-sequence assembly errors under the large distance constraint [42].

3 Algorithm Description

3.1 Multi-Verse Optimizer (MVO) algorithm

The inspiration for the Multi-Verse Optimizer (MVO) algorithm comes from multiverse theory. In the multiverse, multiple universes attract or collide with each other. Black holes, white holes, and wormholes can be generated in each universe. Multiple universes can exchange matter through black holes and white wormholes in order to achieve a stable state. Since each universe is continually expanding, it has an expansion rate. These characteristics of the multiverse inspired Mirjalili [43] to propose the MVO algorithm. The mechanism of the MVO algorithm is that matter is transferred from low-fitness universes to high-fitness universes via channels. This process increases the average fitness value and reaches a steady state.

In the MVO algorithm [43], the current universe is first sorted according to the standardized expansion rate, and roulette is then used during each iteration in order to determine which universe contains the white hole. The search space is explored using the black hole and white hole mechanism. As the rate of expansion of the universe increases, the probability of successfully transferring objects increases. Wormholes can appear randomly in all universes without considering expansion rates. In order to better explore the universe, assume that wormholes are frequently established in the current universe and the best-fitness universe. The specific expression of the wormhole is as follows:

$$x_{ij} = \begin{cases} X_j + TDR \times ((ub_j - lb_j) \times r4 + lb_j), r3 < 0.5 & r2 < WEP \\ X_j - TDR \times ((ub_j - lb_j) \times r4 + lb_j), r3 \ge 0.5 & \\ x_{ij} & r2 \ge WEP \end{cases} \tag{6}$$

where X_j indicates the *jth* variable of best universe formed so far, travelling distance rate (*TDR*) and wormhole existence probability (*WEP*) is coefficient, lb_j and ub_j is the lower/upper bound of jth parameter, x_{ij} indicates the *jth* parameter of ith universe, and *r2, r3, r4* are random numbers in [0, 1]. More details are consistent with Mirjalili's paper [43].

$$WEP = \min + (\max - \min) \times \frac{l}{L} \tag{7}$$

where *min* and *max* is the minimum/maximum, and values are assigned to 0.2 and 1 respectively in this manuscripts, l shows the current iteration, and L indicates the maximum iterations.

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{8}$$

where p defines the exploitation accuracy over the iterations, and in this paper p= *6*. The higher the p value, the faster and more accurate the local search.

3.2 Damp Multi-Verse Optimizer (DMVO) algorithm

In the MVO algorithm, the individual update mainly depends on the inflation rate, and the population evolution is based on the current optimal global wormhole existence probability, *WEP*. Because the early optimal value of the algorithm is often too far from the actual value, the effect of this strategy in the early stage is not ideal. The MVO algorithm uses the traveling distance rate (*TDR)* as the convergence factor, where p defines the accuracy of the iterative development. The convergence curve of *TDR*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

is shown in Figure 1. Although the quadratic convergence curve is better than the straight line, the lack of disturbance during the search iteration process will increase the probability of the algorithm falling into a local optimum, resulting in the algorithm's stalled optimization. The convergence factor *TDR* was improved by adding the disturbance factor *C1*; the convergence curve after the addition is also shown in Figure 1. The DMVO flowchart and pseudo-code are shown in Figure 2 and Algorithm 1.

A study by Reynolds [44] revealed that fruit flies explore their environment through a series of straight flight paths interspersed with sudden 90-degree turns. This research proposed an intermittent scale-free search mode, which is known as Lévy flight. Subsequently, this model was applied to optimization and optimal search, and preliminary results have indicated that it performs well [45]. In nature, animals look for food in a random or quasi-random manner. In the multiverse, the universe where the next black/white hole appears can also be approximated as a black/white hole looking for food (universes). The update strategy of Lévy flight is applied in the DMVO algorithm instead of the random update based on the current global optimum, thereby reducing the influence of individuals with local minimum or local maximum values on the update mechanism.

The damping model was originally established in materials science in order to handle the complex internal structure of damping materials. In recent years, the damping factor in the damping model has been introduced by researchers into the fields of engineering, structural mechanics, machine learning, and other disciplines. In addition, some researchers have used damping factor optimization algorithms [46]. Inspired by the damping model, the damping factor is introduced into the MVO algorithm. The lack of perturbations in the *TDR* of the MVO algorithm may lead to a lack of creativity in the search iteration process. To correct this, a damping factor is added to the convergence factor *TDR* in order to increase the disturbance. In Figure 1, the damping factor *C1* is compared with the original *TDR*. It can be seen that *C1* is not a traditional nonlinear curve but rather a convergence curve with oscillation. When an undesirable state is reached in the optimization process, the oscillating effect

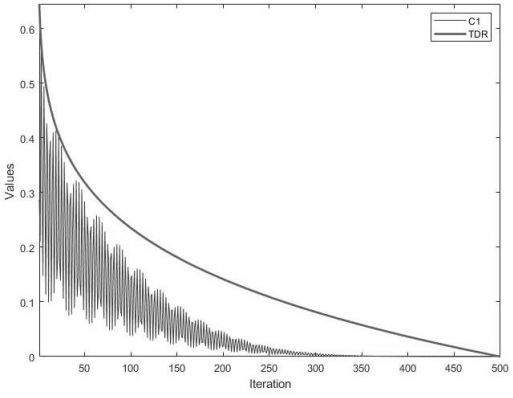


Fig. 1. Comparison of TDR with disturbance factor C1

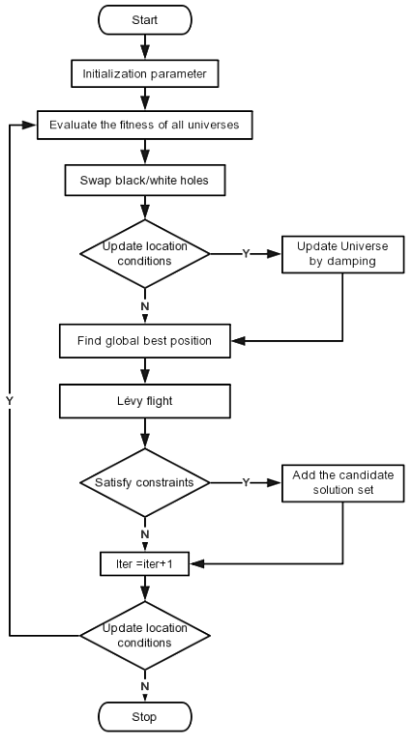


Fig. 2. DMVO algorithm flowchart

Algorithm 1. DMVO 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)+C1*Traveling_
            distance_rate* ((ub(j)- lb(j)) * r4 + lb(j));
            else
                U(i,j)=Best_universe(j)-
            C1*Traveling_distance_rate* ((ub(j)- lb(j)) *
            r4 + lb(j));
            end if
        end if
    end for
    If Time>MaxTime/2
        Buniverse= Lévy * Best_universe;
    end if
end for
```

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

will jump out of the current state and attempt to search a wider range. In this algorithm, an orthogonal damping model for Rayleigh damping is applied. The mathematical expression for Rayleigh damping is

$$C = a_0 M + a_1 K \tag{9}$$

After multiple tests on different test functions, the weights a_0 and a_1 of the damping model were set to 0.225 and 0.4, respectively. In order to further verify the search ability of the algorithm after adding the damping factor, the DMVO algorithm without the damping factor was also tested. Detailed numerical comparisons are provided in Section 4. In order to further clarify the distinction between algorithms, the comparison algorithm without the damping factor is referred to as the DMVO-nodamp. The relevant parameters in formula (10) are consistent with (6) except for *C1*.In (10), *Time* and *Max_Time* represent the current iteration number and the maximum iteration number respectively, and *cos* is the standard cosine function.

$$x_{ij}=\begin{cases}\begin{cases}X_j+TDR*\mathrm{C1}*((ub_j-lb_j)*r4+lb_j), r3<0.5 & r2<WEP\\ X_j-TDR*\mathrm{C1}*((ub_j-lb_j)*r4+lb_j), r3\geq 0.5\end{cases}\\ x_{ij} \qquad r2\geq WEP\end{cases} \tag{10}$$

$$\mathrm{C}_1=(0.225*\mathrm{e}+0.4*\cos(2*\mathrm{Time}))*\mathrm{e}^{-(\frac{8*Time}{3*Max\_time})^2} \tag{11}$$

3.3 Experiment environment

The experiment used a computer with Intel core i7 RAM 8G and MATLAB 2018a software; the results are shown in Tables 2–8; the 4 bases were mapped to the 4 numbers 0–3 in the DNA coding set (*T* → *0*, *C* → *1*, *G* → *2*, *A* → *3*); the values of the parameters related to the DMVO algorithm were consistent with the original MVO study; the key parameters are listed in Table 1.

TABLE 1. PARAMETER VALUES IN WEP AND TDR

Parameter	Value
min	0.2
max	1
p	6

3.4 Benchmark functions

In order to better demonstrate the performance of our improved algorithm, we tested it using mainstream benchmark functions [47–49]. Each heuristic algorithm has particular practical problems that it is good at solving. On the other hand, not every algorithm can achieve optimal results for all problems. The benchmark test function is a simulation of an actual problem; thus, different algorithms may excel at different test functions. Thirteen benchmark test functions were selected, including 7 high-dimensional unimodal functions and 6 high-dimensional multimodal functions. These 13 test functions represent most types of optimization problems, and testing them is therefore a good measure of algorithm performance. In order to improve the credibility of the test results and the feasibility of the experiments, the definition domain of the test functions was limited.

After the 13 test functions were run 30 times, the averages and standard deviations of the results were compared with the previous algorithm and other representative algorithms. We selected the GA [50], PSO [51], GSA [52], GWO [53], MVO [43], and K-means Multi-Verse Optimizer (KMVO) algorithms for comparison. Among them, the GA is the earliest and is a representative evolutionary algorithm, the PSO is a heuristic algorithm that imitates the behavior of groups; the GSA is the best algorithm based on physical theory, and the KMVO is a recent improvement of our previous work. The maximum number of iterations for these algorithms was set to 500. The results of the MVO, GWO, GSA, PSO, and GA algorithms were from the work of Mirjalili [43]. The data for the KMVO were derived from our recent article. The benchmark functions 1–13 were consistent with those in Mirjalili's research [43]. We also performed Wilcoxon's nonparametric rank sum detection and evaluated the results. Due to the randomness of the heuristic algorithm, the running result p of the statistical rank sum is more convincing. When $p > 0.05$, it can be said that the algorithm is statistically significant.

F1–F7 are high-dimensional unimodal functions that have global optima, so they are usually used for the universal testing of algorithms. F8–F13 each have a single global optimum and several local optima. As the dimension increases, the number of local optimum solutions also increases. This increases the difficulty of solving the heuristic algorithm and can better reflect the speed of optimizing an algorithm as well as the performance of

TABLE 2.
AVERAGE RESULTS OF UNIMODAL BENCHMARK FUNCTIONS

F	DMVO	DMVO-nodamp	KMVO	MVO [43]	GWO [43]	GSA [43]	PSO [43]	GA [43]
	Ave	Ave	Ave	Ave	Ave	Ave	Ave	Ave
F1	1.67E-17	1.96E-08	8.8094	2.08583	2319.19	2983.667	3.552364	27,187.58
F2	6.83E-10	5.24E-06	3.1771	15.92479	14.43166	10.96518	8.716272	68.6618
F3	5.75E-17	1.26E-07	2209.7671	453.2002	7278.133	113,740.40	2380.963	48,530.91
F4	2.87E-09	4.46E-05	1	3.123005	13.09729	32.2563	21.5169	62.99326
F5	48.4392	48.956	603.1599	1272.13	3,425,462	7582.498	1132.486	65,361,620
F6	0.17774	6.719	9.3446	2.29495	5009.442	74,617.45	86.62074	49,574.10
F7	1.45E-04	3.95E-04	0.11137	0.051991	0.408082	21.16092	0.577434	18.72524

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

jumping out of the local optimum. The significance of comparing the average and the standard deviation of the test function results stems from the fact that the results of each iteration run of the heuristic algorithm are inconsistent. Therefore, weighing the pros and cons of a heuristic algorithm cannot be based on the results of a single run, whereas averaging the results of 30 runs can eliminate the uncertainty, and calculating the standard deviation of the results can reveal the stability of the result.

4 RESULTS

4.1 High-dimensional unimodal functions

In order to test the performance of the DMVO algorithm in practical applications, it was compared with other algorithms such as the PSO, MVO, and others using test functions. Tables 2 and 3 list the results after 30 runs. It can be seen from these tables that the 2 test indicators (the average and the standard deviation) have been significantly improved in the DMVO. In particular, compared with the KMVO algorithm for F1 and F3, the average (Ave) and the standard deviation (SD) have been improved by more than 10 orders of magnitude. Of course, the DMVO also performs much better than the GWO and PSO algorithms. For the function F4, the standard deviation of the DMVO algorithm is better than that of the KMVO algorithm, although it should be noted that the average value of the KMVO algorithm is 1, which may indicate that the KMVO algorithm has fallen into the local optimal value of 1. The DMVO algorithm, however, jumps out of the local optimum and reaches the order of magnitude e-9, thus illustrating the competitiveness of the DMVO algorithm. Furthermore, the comparison of the DMVO and DMVO-nodamp reveals the necessity of adding damping disturbances. For example, in F6, the DMVO-nodamp ends the search in advance, while the DMVO continues to iteratively obtain a solution close to the true optimal.

4.2 High-dimensional multimodal functions

Tables 4 and 5 list the results after 30 runs for functions F7–F13. Compared with high-dimensional unimodal functions, high-dimensional multimodal functions have multiple local optimal solutions. Heuristic algorithms are thus more likely to fall into local optima during the optimization process, which is an excellent test for the ability of heuristic algorithms to jump out of local optima. Compared with the recently improved KMVO algorithm, the average and the standard deviation of the DMVO algorithm for F10 have been improved by 10 orders of magnitude, which illustrate the necessity of adding the damping factor and Lévy flight to the DMVO algorithm. For functions F9 and F11, the DMVO algorithm found the global optimal value of the test function after 500 iterations, further illustrating the optimizing performance of the DMVO algorithm and its ability to jump out of the local optimum. In all of the multi-modal benchmark functions, it can clearly be seen that the DMVO algorithm exhibits significant improvements in both Ave and SD,

TABLE 3.
SD RESULTS OF UNIMODAL BENCHMARK FUNCTIONS

F	DMVO	DMVO-nodamp	KMVO	MVO [43]	GWO [43]	GSA [43]	PSO [43]	GA [43]
	SD	SD	SD	SD	SD	SD	SD	SD
F1	1.87E-17	5.37E-08	1.6306	0.648651	1237.109	903.3827	2.853733	2745.82
F2	3.80E-10	7.07E-06	0.72651	44.7459	5.923015	10.54968	4.929157	6.062311
F3	5.08E-17	3.41E-07	786.9794	177.0973	2143.116	78,786.15	1183.351	8249.75
F4	1.37E-09	8.47E-05	0	1.582907	11.3469	6.226765	6.71628	2.535643
F5	0.081404	9.26E-02	611.1058	1479.477	3,304,309	7314.818	1357.967	29,714,021
F6	0.080863	9.93E-01	2.3629	0.630813	3028.875	8231.224	147.3067	8545.149
F7	1.61E-04	2.65E-04	0.028805	0.029606	0.119544	12.1566	0.318544	4.935256

TABLE 4.
AVERAGE RESULTS OF MULTI-MODAL BENCHMARK FUNCTIONS

F	DMVO	DMVO-nodamp	KMVO	MVO [43]	GWO [43]	GSA [43]	PSO [43]
	Ave	Ave	Ave	Ave	Ave	Ave	Ave
F8	−12473.8746	−1.20E+04	−12,348.6192	−11,720.20	-10,739.50	-4638.41	-6727.59
F9	0	7.60E−10	49.9577	118.046	89.13475	128.0103	99.83202
F10	5.89E−10	5.94E−06	2.9395	4.074904	9.452571	1.654073	4.295044
F11	0	8.19E−09	0.75229	0.938733	22.51942	1021.705	624.3092
F12	0.073642	2.70E−01	1.898	2.459953	3,200,008	741,596.90	13.38384
F13	4.68E+00	4.7621	1.35E−32	0.222672	7,815,082	6,670,046	21.11298

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 5.
SD RESULTS OF MULTI-MODAL BENCHMARK FUNCTIONS

F	DMVO SD	DMVO-nodamp SD	KMVO SD	MVO [43] SD	GWO [43] SD	GSA [43] SD	PSO[43] SD
F8	693.532	9.87E+02	724.6721	937.1975	1162.793	805.0488	1352.882
F9	0	2.39E−09	0.017879	39.34364	37.95765	26.90054	24.62872
F10	2.53E-10	1.16E−05	0.57575	5.501546	3.467608	1.583499	1.308386
F11	0	2.64E−08	0.049873	0.059535	26.68168	82.95486	105.3874
F12	0.057076	7.07E−02	0.61601	0.791886	6,746,208	624,375.50	8.969122
F13	5.38E-02	6.54E−02	5.57E−48	0.086407	16,475,640	5,719,826	12.83179

again confirming the necessity of adding the damping factor when tackling practical problems. For F13, our algorithm did not perform as well as the KMVO. As mentioned above, no algorithm can achieve optimal for all problems. This unsatisfactory result may be due to the periodicity in the test function F13.

4.3 Wilcoxon's non-parametric rank-sum test

Rank sum testing is an alternative method that can be used when the assumption of the distribution is doubtful. The Wilcoxon rank-sum test [54] is a non-parametric alternative to the 2-sample *t* test, which is based entirely on the ranking of 2 samples. If the results of the Wilcoxon rank-sum test reject the null hypothesis, the results are considered statistically significant. The rank-sum of any 2 of the 30 rounds of operations was calculated, and the result of the hypothesis test was obtained by comparing the rank sums pair by pair. When $p > 0.05$, this indicates that p rejected the null hypothesis, proving that the results are highly competitive. As shown in Table 6, the results met the $p > 0.05$ criterion in most cases, thus illustrating the statistical significance of the DMVO algorithm.

TABLE 6.
P VALUES OF WILCOXON RANK SUM TEST OVER 30 RUNS

F	DMVO	KMVO	MVO[43]	GWO[43]	GSA[43]
F1	0.52161	0.34817	N/A	0.002827	0.000183
F2	0.48536	0.51033	0.009108	0.053903	0.909722
F3	0.50454	0.87769	N/A	0.000183	0.000183
F4	0.49386	N/A	N/A	0.140465	0.000183
F5	0.52008	N/A	0.677585	0.10411	0.005795
F6	0.54341	0.58492	N/A	0.000183	0.000182
F7	0.47454	0.62052	N/A	0.000183	0.000183
F8	0.51805	0.60117	N/A	0.053903	0.000183
F9	0.49819	0.26379	0.121225	N/A	0.002827
F10	0.47312	0.35904	0.121225	0.001315	N/A
F11	0.46246	0.59211	N/A	0.005795	0.000183
F12	0.47573	0.48449	N/A	0.025748	0.000183
F13	0.4587	N/A	N/A	0.075662	0.000183

4.4 Bounds on DNA storage codes

Initiated by Joe Davis, the "Microvenus project" stores non-biological data such as images and audio in DNA and encodes them based on the size of the molecule *CTAG* (*C*→1, *T*→2, *A*→3, *G*→4) [5], for example, 10101 → *CCCC*, 100101 → *CTCCT*. However, because the decoding result is not unique, it may cause errors. In 2012, Church et al. [10] used a free base swap strategy (*0*→*A* or *T*, 1→*C* or *G*), and they completed 0.65 *MB* of data into 8.8 *MB* of DNA oligonucleotides (*nt*) with a length of 159 nucleotides (*nt*). Similarly, Two bytes (16 binary bits) are mapped to 9 bases and eight binary bits are mapped to 5 bases. This type of method has no corresponding consideration for errors in the practical application of DNA synthesis and sequencing. Soon after, some researchers used Huffman codes [11] and error-correcting codes to correct the errors of synthesis and sequencing. But for the no-payload bit, it is not worth adding error-correcting redundancy due to the short length.

The steps of DMVO algorithm to construct DNA storage coding set are as follows:

Step 1: Generate the initial universe population and initialize the parameters required by the algorithm. Generating candidate coding, the candidate coding originate from possible DNA coding based on the combinatorial constraints;

Step 2: The DNA storage coding set is initialized, the initial universe population is sorted by the wormhole strategy in the MVO algorithm, and the currently optimal fitness universe is added to the candidate solution set of the universe;

Step 3: Through the white hole and black hole tunnels strategy to select the universe and exchange of objects, through the formulas (9), (10), (11) to update universe;

Step 4: The updated results were used as the input of Lévy flight search for the best fitness universe;

Step 5: Determine whether the combination constraint is satisfied. If the combination constraint is satisfied, add the new DNA coding to the DNA storage coding set;

Step 6: Completing the number of iterations and output the DNA storage coding set;

$A^{GC,NL,UA}(n,d,w)$ is defined as a DNA storage coding set of length n and edit distance d, which satisfies the distance constraint, *GC* content constraint, No-runlength constraint, and uncorrelated address constraint. In Tables 7 and 8, d represents the Hamming distance and the storage editing distance, respectively. The values in Tables 7 and 8 represent the lower bounds when $4 \le n \le 10,\ 3 \le d \le n$. The bold values represent cases in which our lower bound is better than that of the previous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2020.3011582, IEEE/ACM Transactions on Computational Biology and Bioinformatics

algorithms (the Altruism algorithm and the KMVO algorithm). Altruistic algorithm is an intelligent algorithm that iteratively deletes potential code words based on greedy algorithm. It deletes the "worst" candidate code words in each iteration. As the algorithm progresses, the altruistic algorithm is used to greedily delete the largest code word within the range of *d-1* until the code book has the minimum distance *d*. The KMVO algorithm, which comes from our previous work, is an improvement of MVO algorithm based on k-means and has the advantages of fast convergence and high population complexity. In KMVO algorithm, the universe is clustered into the optimal class and the worst class after each update of the position of the universe. Then, through the wormhole cross strategy, the result is entered into the next iteration together with the optimal class. The coding set constructed by KMVO algorithm has better code boundary than the altruistic algorithm which satisfies the same constraint. The superscripts are identified in Table 9. A comparison of the lower bounds obtained by the DMVO algorithm with the previous best results revealed that most of them were better than the previous results. The results showed that the size of the DNA storage coding set obtained by the DMVO algorithm increased by 4–11%. For example, when $n = 9$ and $d = 4$, the result obtained by the DMVO algorithm was 10% higher than the previous optimal result.

TABLE 7. LOWER BOUNDS FOR $A^{GC,NL}(n,d,w)$

n\d	3	4	5	6	7	8	9	10
4	12^{k}							
	12^{u}							
5	20^{k}	8^{k}						
	20^{u}	**8^{u}**						
6	56^{k}	23^{k}	$8^{k,a}$					
	58^{u}	**24^{u}**	8^{u}					
7	**127^{k}**	**45^{k}**	16^{k}	6^{k}				
	125^{u}	44^{u}	**17^{u}**	**7^{u}**				
8	319^{k}	94^{k}	32^{k}	13^{k}	5^{k}	$4^{k,a}$		
	324^{u}	**106^{u}**	**35^{u}**	**14^{u}**	**5^{u}**	**4^{u}**		
9	680^{k}	202^{k}	**65^{k}**	23^{k}	10^{k}	5^{k}	$4^{k,a}$	
	713^{u}	**223^{u}**	64^{u}	**24^{u}**	**10^{u}**	**5^{u}**	**4^{u}**	
10	**2081^{k}**	547^{k}	151^{k}	**54^{k}**	19^{k}	9^{k}	$4^{k,a}$	
	1906^{u}	**555^{u}**	**159^{u}**	51^{u}	**20^{u}**	**10^{u}**	**4^{u}**	

TABLE 9. MEANINGS OF SUPERSCRIPTS

Superscript	Meaning
a	Altruistic algorithm
k	KMVO algorithm
u	DMVO algorithm

dress constraint. It can be seen from Table 8 that the DNA coding set constructed by the DMVO is larger than the corresponding set constructed by the KMVO in almost every case. This result confirms the applicability of the DMVO algorithm to practical problems and demonstrates that this method features good optimization performance. The results revealed that the size of the DNA storage coding set obtained by the DMVO algorithm increased by 6–16%. The increase in quantity was not as significant as shown in Table 7, however. This may be because the number of eligible sequences tends to decrease as the number of candidate sets decreases. Therefore, even if a better algorithm is used, only a small number of new DNA sequences that satisfy the constraints can be searched. On the other hand, when the candidate set increases, the DMVO algorithm can also find a relatively large number of sequences. For example, when $n = 10$ and $d = 3$, the DMVO can also provide a very effective improvement. At $n = 7$, $d = 5$ or 6, the number of coding sets approached the accurate value, and both algorithms obtained the same results.

TABLE 8. LOWER BOUNDS FOR $A^{GC,NL,UA}(n,d,w)$

n\d	3	4	5	6	7	8	9	10
4	5^{k}							
	6^{u}							
5	12^{k}	5						
	12^{u}	**5^{u}**						
6	30^{k}	10^{k}	4^{k}					
	30^{u}	**11^{u}**	**4^{u}**					
7	50^{k}	18^{k}	6^{k}	3^{k}				
	53^{u}	**19^{u}**	**6^{u}**	**3^{u}**				
8	95^{k}	37^{k}	13^{k}	4^{k}	3^{k}			
	101^{u}	**38^{u}**	12^{u}	**5^{u}**	**3^{u}**			
9	155^{k}	55^{k}	19^{k}	7^{k}	3^{k}	2^{k}		
	167^{u}	**58^{u}**	**19^{u}**	**7^{u}**	**3^{u}**	**2^{u}**		
10	227^{k}	95^{k}	32^{k}	10^{k}	5^{k}	3^{k}	2^{k}	
	250^{u}	**110^{u}**	**34^{u}**	**11^{u}**	**5^{u}**	**3^{u}**	**2^{u}**	

In order to illustrate the effectiveness of the storage edit distance constraint, a thermodynamic analysis experiment was performed. Specifically, the variance of the

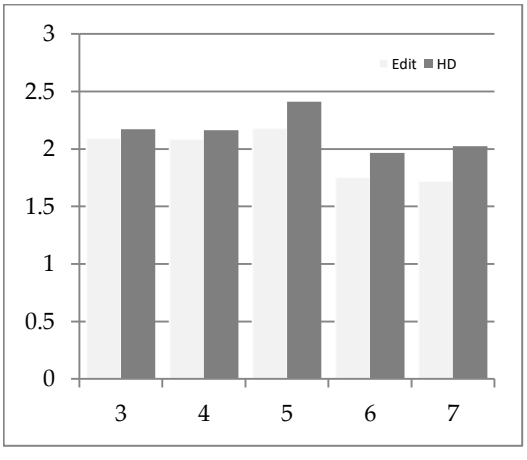


Fig. 3. Tm under 2 conditions when n = 9

Similarly, the lower bounds of the DNA storage coding set were compared when the distance constraint was the edit distance. $A^{GC,NL,UA}(n,d,w)$ was defined to represent the DNA storage coding set for a given length *n*, and the storage edit distance was *d*, satisfying the *GC* content constraint, No-runlength constraint, and uncorrelated ad-

TABLE 10. THERMODYNAMIC TM VARIANCE COMPARISONS

n\d		3	4	5	6	7
8	Edit	**2.1847**	**2.3382**	**1.7156**	**2.2094**	**0.5924**
	HD	2.3701	2.4229	2.9256	2.3299	2.4566
9	Edit	**2.0953**	**2.0851**	**2.1794**	**1.7534**	**1.7198**
	HD	2.173	2.1646	2.4109	1.9665	2.0243
10	Edit	**2.0879**	**1.9431**	**2.1939**	**1.5834**	2.4125
	HD	2.1039	2.1165	2.2283	2.2036	**2.164**

melting temperatures in the 2 DNA coding sets was calculated. The melting temperature (Tm) [55] is the temperature at which double-stranded DNA molecules become single-stranded during DNA denaturation.

In this study, the composition of the DNA sequence was the primary focus, and the given concentration of the DNA molecules was 10 *nM*, while the salt concentration was 1 *M*. A thermodynamic analysis was performed on the DNA coding set obtained by the DMVO algorithm under the Hamming distance constraint and the storage edit distance constraint. It can be seen from Table 10 that in most cases, a coding set with a small variance of Tm could be obtained by the storage edit distance constraint. In particular, Figure 3 is a bar chart illustrating the comparison when $n = 9$, where the abscissa represents the distance d ($2 < d < 8$), and the ordinate represents the variance of the Tm. As can be seen from this figure, for the case in which n and d are the same, the coding set obtained by the edit distance in the DMVO algorithm is smaller than the Tm variance of the coding set obtained by the Hamming distance. In order to ensure fairness, a thermodynamic comparison was also made with the results listed by Limbachiya [39]. When $n = 8$ and $d = 3$, the variance of Tm the coding sets obtained by the altruistic algorithm under the Hamming distance constraint was 2.3733, while the result in Table 10 was 2.1847.

It can be seen from Tables 7–10 that the size of the DNA storage coding set obtained by the DMVO algorithm increased by 4–16%, while the Tm variance decreased by 3–18%. A larger coding set was constructed to illustrate the excellent performance of the DMVO algorithm in terms of practical applications. In DNA storage, larger non-payload sets act as address bits or primers for DNA sequences, meaning that more DNA sequences can be indexed by shorter addresses. When the variance of the coding set Tm with the storage edit distance constraint is smaller, the PCR reaction is more stable, again proving the applicability of the storage edit distance. In other aspects, computational models such as spiking neural networks [56], reinforcement learning, and parallel computing models [57], which are similar to neurons, can be considered for coding set design applied to DNA.

5 CONCLUSIONS

Molecular biologists have pointed out that nanopore sequencing is prone to addition, deletion, and misreading errors. For this reason, we introduced the storage distance constraint, which is similar to the nanopore sequencing error situation. In addition, an uncorrelated address constraint was proposed in order to avoid misreading address bits, and a comparison of address bits with and without the uncorrelated address constraint was presented. Moreover, the applicability of the 2 new constraints was proven via thermodynamic comparison experiments. In the analysis and conclusion, the DMVO algorithm under 2 different combination constraints was compared with the DNA storage coding set obtained by the previous algorithm. In order to illustrate the superiority of the storage edit distance constraint and the uncorrelated address constraint, the Tm variance of the DNA storage coding set was compared. The results presented in Tables 7–10 revealed that the size of the DNA storage coding set obtained by the DMVO algorithm increased by 4–16%, while the Tm variance decreased by 3–18%.

The proposed DMVO algorithm constructs a non-payload coding set for DNA storage. In this study, the combination constraints in the DNA storage coding problem were abstracted as a multi-objective optimization problem, and the heuristic algorithm was employed to determine the approximate optimal solution of the DNA storage coding problem. The DMVO algorithm was inspired by the damping factor and Lévy flight. These 2 strategies were used to improve the MVO algorithm when updating the next generation population in a single way, as well as the problem of falling into a local optimum during the late optimization stage. The DMVO algorithm was compared with the KMVO, MVO, GA, GSA, PSO, and other algorithms via the quantitative comparison (Ave and SD) of the benchmark functions, displaying relatively good results, as listed in Tables 7–10.

In future work, we will continue to focus on the coding of DNA storage. As the constraints increase, the DNA storage coding set will become smaller, i.e., the higher the quality of the coding, the smaller the coding set will become. Finding a balance between the size of the DNA coding set and coding set quality is a conundrum we need to ponder. In addition, we will fine-tune our algorithms in order to obtain better results and apply them to the field of DNA storage coding or the solution of other engineering problems.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (No. 2018YFC0910500), the National Natural Science Foundation of China (Nos.61425002, 61751203, 61772100, 61972266, 61802040, 61672121, 61572093), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT_15R07), the Program for Liaoning Innovative Research Team in University(No.LT2017012), the Natural Science Foundation of Liaoning Province (Nos. 20180551241, 2019-ZD-0567), the High-level Talent Innovation Support Program of Dalian City (Nos.2017RQ060, 2018RQ75), the Dalian Outstanding Young Science and Technology Talent Support Program (No.2017RJ08), Scientific Research Fund of Liaoning Provincial Education Department (No. JYT19051). Bin Wang and Qiang Zhang are the corresponding authors of this paper.

REFERENCES

[1] A. Siddiqa, A. Karim, and A. Gani, "Big data storage technologies: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 8, pp. 1040-1070, Aug, 2017.

[2] H. N. Poinar, D. Serre, V. Jaenickedespr, J. Hebler, N. Rohland, M. Kuch, J. Krause, L. Vigilant, and M. Hofreiter, "Genetic Analyses from Ancient DNA," *Annual Review of Genetics*, vol. 38, no. 1, pp. 645-679, 2004.

[3] Y. Erlich, and D. Zielinski, "DNA Fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950-953, Mar, 2017.

[4] N. MS, "Some fundamental issues of microminiaturization," *Radiotekhnika*, no. No.1, pp. 3–12, 1964.

[5] D. J, "Microvenus. ," Art J vol. 55:70–74, 1996.

[6] C. Bancroft, T. Bowler, B. Bloom, and C. Clelland, "Long-Term Storage of Information in DNA," *Science*, vol. 293, no. 5536, pp. 1763-1765, 2001.

[7] M. Arita, and Y. Ohashi, "Secret Signatures Inside Genomic DNA," *Biotechnology Progress*, vol. 20, no. 5, pp. 1605-1607, 2004.

[8] P. C. Wong, K. K. Wong, and H. P. Foote, "Organic data memory using the DNA approach," *Communications of The ACM*, vol. 46, no. 1, pp. 95-98, 2003.

[9] H. H. Nguyen, J. Park, S. J. Park, C. S. Lee, S. Hwang, Y. B. Shin, T. H. Ha, and M. Kim, "Long-Term Stability and Integrity of Plasmid-Based DNA Data Storage," *Polymers*, vol. 10, no. 1, Jan, 2018.

[10] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628-1628, 2012.

[11] N. M. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. Leproust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77-80, 2013.

[12] W. D. Chen, A. X. Kohll, B. H. Nguyen, J. Koch, R. Heckel, W. J. Stark, L. Ceze, K. Strauss, and R. N. Grass, "Combining Data Longevity with High Storage Capacity-Layer-by-Layer DNA Encapsulated in Magnetic Nanoparticles," *Advanced Functional Materials*, vol. 29, no. 28, Jul, 2019.

[13] Y. Wang, M. Keith, A. Leyme, S. Bergelson, and M. Feschenko, "Monitoring long-term DNA storage via absolute copy number quantification by ddPCR," *Analytical biochemistry*, vol. 583, pp. 113363, 2019 Oct 15 (Epub 2019 Jul, 2019.

[14] R. Heckel, G. Mikutis, and R. N. Grass, "A Characterization of the DNA Data Storage Channel," *Scientific Reports*, vol. 9, Jul, 2019.

[15] D. Carmean, L. Ceze, G. Seelig, K. Stewart, K. Strauss, and M. Willsey, "DNA Data Storage and Hybrid Molecular-Electronic Computing," *Proceedings of the Ieee*, vol. 107, no. 1, pp. 63-72, Jan, 2019.

[16] L. C. Meiser, P. L. Antkowiak, J. Koch, W. D. Chen, A. X. Kohll, W. J. Stark, R. Heckel, and R. N. Grass, "Reading and writing digital data in DNA," *Nature Protocols*, vol. 15, no. 1, pp. 86-101, 2020.

[17] Z. Ping, D. Z. Ma, X. L. Huang, S. H. Chen, L. Y. Liu, F. Guo, S. J. Zhu, and Y. Shen, "Carbon-based archiving: current progress and future prospects of DNA-based data storage," *Gigascience*, vol. 8, no. 6, Jun, 2019.

[18] K. O. Cheng, N. F. Law, and W. C. Siu, "Clustering-Based Compression for Population DNA Sequences," *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 208-221, Jan-Feb, 2019.

[19] T. Pan, P. Flick, C. Jain, Y. C. Liu, and S. Aluru, "Kmerind: A Flexible Parallel Library for K-mer Indexing of Biological Sequences on Distributed Memory Systems," *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 4, pp. 1117-1131, Jul-Aug, 2019.

[20] M. Jain, S. Koren, K. H. Miga, J. Quick, A. C. Rand, T. A. Sasani, J. R. Tyson, A. D. Beggs, A. Dilthey, and I. T. Fiddes, "Nanopore sequencing and assembly of a human genome with ultra-long reads," *Nature Biotechnology*, vol. 36, no. 4, pp. 338-345, 2018.

[21] A. Magi, R. Semeraro, A. Mingrino, B. Giusti, and R. Daurizio, "Nanopore sequencing data analysis: state of the art, applications and challenges," *Briefings in Bioinformatics*, vol. 19, no. 6, pp. 1256-1272, 2017.

[22] B. M. Venkatesan, and R. Bashir, "Nanopore sensors for nucleic acid analysis," *Nature Nanotechnology*, vol. 6, no. 10, pp. 615-624, 2011.

[23] M. H. Schmidt, A. Vogel, A. K. Denton, B. Istace, A. Wormit, H. V. De Geest, M. E. Bolger, S. Alseekh, J. Mas, and C. Pfaff, "De Novo Assembly of a New Solanum pennellii Accession Using Nanopore Sequencing," *The Plant Cell*, vol. 29, no. 10, pp. 2336-2348, 2017.

[24] M. H. Garzon, and R. Deaton, "Codeword design and information encoding in DNA ensembles," *Natural Computing*, vol. 3, no. 3, pp. 253-292, 2004.

[25] 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," *Angewandte Chemie*, vol. 54, no. 8, pp. 2552-2555, 2015.

[26] M. Blawat, K. Gaedke, I. Hutter, X. Chen, B. M. Turczyk, S. A. Inverso, B. W. Pruitt, and G. M. Church, "Forward Error Correction for DNA Data Storage," *international conference on conceptual structures*, vol. 80, no. 80, pp. 1011-1022, 2016.

[27] R. Gabrys, H. M. Kiah, and O. Milenkovic, "Asymmetric Lee Distance Codes for DNA-Based Storage," I*eee Transactions on Information Theory*, vol. 63, no. 8, pp. 4982-4995, Aug, 2017.

[28] W. T. Song, K. Cai, M. Zhang, and C. Yuen, "Codes With Run-Length and GC-Content Constraints for DNA-Based Data Storage," *Ieee Communications Letters*, vol. 22, no. 10, pp. 2004-2007, Oct, 2018.

[29] K. A. S. Immink, and K. Cai, "Design of Capacity-Approaching Constrained Codes for DNA-Based Storage Systems," *Ieee Communications Letters*, vol. 22, no. 2, pp. 224-227, Feb, 2018.

[30] S. Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, "Mutually Uncorrelated Primers for DNA-Based Data Storage," *Ieee Transactions on Information Theory*, vol. 64, no. 9, pp. 6283-6296, Sep, 2018.

[31] Y. X. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, and C. L. Poh, "Construction of Bio-Constrained Code for DNA Data Storage," *Ieee Communications Letters*, vol. 23, no. 6, pp. 963-966, Jun, 2019.

[32] L. Song, and A. Zeng, "Orthogonal Information Encoding in Living Cells with High Error-Tolerance, Safety, and Fidelity," *ACS Synthetic Biology*, vol. 7, no. 3, pp. 866-874, 2018.

[33] 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 Communications*, vol. 10, Jun, 2019.

[34] Q. Yin, B. Cao, X. Li, B. Wang, Q. Zhang, and X. Wei, "An Intelligent Optimization Algorithm for Constructing a DNA Storage Code: NOL-HHO," *International journal of molecular sciences*, vol. 21, no. 6, 2020 Mar, 2020.

[35] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using DNA," *Nature Reviews Genetics*, vol. 20, no. 8, pp. 1, 2019.

[36] D. H. Smith, N. Aboluion, R. Montemanni, and S. Perkins, "Linear and nonlinear constructions of DNA codes with Hamming distance d and constant GC-content," *Discrete Mathematics*, vol. 311, no. 13, pp. 1207-1219, 2011.

[37] B. Wang, X. D. Zheng, S. H. Zhou, C. J. Zhou, X. P. Wei, Q. Zhang, and Z. Q. Wei, "Constructing DNA Barcode Sets Based on Particle Swarm Optimization," *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 3, pp. 999-1002, May-Jun, 2018.

[38] B. Wang, Q. Zhang, and X. Wei, "Tabu Variable Neighborhood Search for Designing DNA Barcodes," *IEEE Transactions on Nanobioscience*, 2020, 19(1): pp. 127-131.

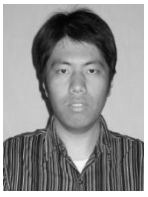
[39] D. Limbachiya, M. K. Gupta, and V. Aggarwal, "Family of Constrained Codes for Archival DNA Data Storage," *IEEE Communications Letters*, vol. 22, no. 10, pp. 1972-1975, 2018.

[40] S. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and Error-Free DNA-Based Data Storage," *Scientific Reports*, vol. 7, Jul, 2017.
[41] M. Levy, and E. Yaakobi, "Mutually Uncorrelated Codes for DNA Storage," *arXiv: Information Theory*, 2018.
[42] S. Yazdi, Y. B. Yuan, J. Ma, H. M. Zhao, and O. Milenkovic, "A Rewritable, Random-Access DNA-Based Storage System," *Scientific Reports*, vol. 5, Sep, 2015.
[43] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing & Applications*, vol. 27, no. 2, pp. 495-513, Feb, 2016.
[44] A. M. Reynolds, and M. A. Frye, "Free-Flight Odor Tracking in Drosophila Is Consistent with an Optimal Intermittent Scale-Free Search," *PLOS ONE*, vol. 2, no. 4, 2007.
[45] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. D. Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, vol. 401, no. 6756, pp. 911-914, 1999.
[46] R. Yu, Z. Yuan, M. Zhao, M. Yu, and X. Lu, "Damping Based Traffic Allocation in Wireless Machine-to-Machine Communications Networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, pp. 814267, 2013.
[47] X. Yao, Y. H. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, 1999.
[48] J. G. Digalakis, and K. G. Margaritis, "ON BENCHMARKING FUNCTIONS FOR GENETIC ALGORITHMS," *International Journal of Computer Mathematics*, vol. 77, no. 4, pp. 481-506, 2001.
[49] X. Yang, "Test Problems in Optimization," arXiv: Optimization and Control, 2010.
[50] R.Deaton, "Genetic search of reliable encodings for DNA based computation," Late-Breaking Papers at the First Conference on Genetic Programming, 1996.
[51] 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 Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171-195, Apr, 2008.
[52] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, Jun, 2009.
[53] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, Mar, 2014.
[54] D. H. Kim, and Y. C. Kim, "Wilcoxon signed rank test using ranked-set sample," *Journal of Applied Mathematics and Computing*, 1996.
[55] D. Y. Lando, A. S. Fridman, C. Chang, I. E. Grigoryan, E. N. Galyuk, O. N. Murashko, C. Chen, and C. Hu, "Determination of melting temperature and temperature melting range for DNA with multi-peak differential melting curves," *Analytical Biochemistry*, vol. 479, pp. 28-36, 2015.
[56] T. Song, X. Zeng, P. Zheng, M. Jiang, and A. Rodríguez-Patón, "A parallel workflow pattern modeling using spiking neural p systems with colored spikes," *IEEE transactions on nanobioscience*, vol. 17, no. 4, pp. 474-484, 2018.
[57] T. Song, L. Pan, T. Wu, P. Zheng, M. L. D. Wong, and A. Rodriguezpaton, "Spiking Neural P Systems With Learning Functions," *IEEE Transactions on Nanobioscience*, vol. 18, no. 2, pp. 176-190, 2019.

Ben Cao was born in Suzhou City, Anhui Province in 1997. He received the B.S. degree in computer science and technology from Huaibei Normal University in Anhui Province in 2018. Now graduate students are pursuing computer science and technology at Dalian University and are working hard to pursue a master's degree. Current research interests are mainly intelligent algorithms, DNA storage and DNA coding optimization.

Ii Xue received a B.S degree in engineering from Jining Medical College in 2018. Currently, she is pursuing a master's degree in the key laboratory of advanced design and intelligent computing in Dalian University, majoring in computer science and technology. And she is engaged in the direction of DNA coding optimization.

XiaoKang Zhang received the B.S. degree from Dezhou College in Shandong Province in 2018. He is pursuing for a master's degree in the Key Laboratory of Advanced Design and Intelligent Computing of Dalian University, majoring in computer technology. Currently his focused on research in the direction of DNA computing.



Bin Wang received the B.S degree in computer science and technology from Dalian University, in June 2006 and the PhD degree in mechanical design and theory from Dalian University of Technology in October 2013. He is an associate professor at Dalian University, and research areas include Intelligence Computing, DNA Sequence Design, DNA Cryptography and Biological Network. So far, he has (co-) authored about 61 papers published.

Qiang Zhang received the B.E. degree in Electronic Engineering from the School of electronic Engineering, Xidian University in 1994. And he received the M.E. degree and the Ph.D. degree in Circuits and Systems from the School of electronic Engineering, Xidian University in 1999 and 2002, respectively. He was a post doctorate in School of Mechanical Engineering, Dalian University of Technology, Dalian. In January 2003, he joined the Liaoning Key Lab of Intelligent Information Processing at Dalian University as an associate professor, where he is currently a Professor. His current research interests are neural networks, artificial intelligence, DNA Computing and information hiding. He has published more than 90 papers in these areas. Dr. Zhang has been a board member of the Engineering Graphics Society Liaoning and an academic committee member of CCF Young Computer Scientists & Engineers Forum Dalian. He won the outstanding science and technology worker award by Liaoning Association for Science and Technology in 2004. He currently serves as an Associate Editor of Journal of Nonlinear Analysis & Applied Mathematics, Journal of Neural Computing Systems, International Journal: Mathematical Manuscripts, International Journal of Hybrid Information Technology, The Open Automation and Control Systems Journal and Journal of Mathematical Sciences: Advances and Applications. He is the Editor-in-Chief of The Open Electrical

Xiaopeng Wei received the B.E. degree in Mechanical Engineering from the School of Mechanical Engineering, Dalian University of Technology, Dalian in 1982. And he received the M.E. degree and the Ph.D degree in CAD&CG from the School of

Mechanical Engineering, Dalian University of Technology, Dalian in 1986 and 1993, respectively. He was a post doctorate in School of Mechanical Engineering, Dalian University of Technology, Dalian from 1993 to 1995. In September 1982, he joined the Dalian University of Technology as a teaching assistant, where he is currently a Professor. From February 1995 to June 1995, he was a visiting scholar in University of Hong kong, Hong kong. From July 1997 to August 1997, he was a visiting professor in Alberta University, Canada. From April 1999 to August 1999, he was a visiting professor in Queensland University of Technology, Australia. His current research interests are neural networks, artificial intelligence, DNA Computing, Mechanical Design and CAD&CG. He has published more than 150 papers in these areas