Enhancing Physical and Thermodynamic Properties of DNA Storage Sets With End-Constraint

Jieqiong Wu, Yanfen Zheng, Bin Wang, and Qiang Zhang

***Abstract*— With the explosion of data, DNA is considered as an ideal carrier for storage due to its high storage density. However, low-quality DNA sets hamper the widespread use of DNA storage. This work proposes a new method to design high-quality DNA storage sets. Firstly, random switch and double-weight offspring strategies are introduced in Double-strategy Black Widow Optimization Algorithm (DBWO). Experimental results of 26 benchmark functions show that the exploration and exploitation abilities of DBWO are greatly improved from previous work. Secondly, DBWO is applied in designing DNA storage sets, and compared with previous work, the lower bounds of storage sets are boosted by 9%–37%. Finally, to improve the poor stabilities of sequences, the End-constraint is proposed in designing DNA storage sets. By measuring the number of hairpin structures, melting temperature, and minimum free energy, it is evaluated that with our innovative constraint, DBWO can construct not only a larger number of storage sets, but also enhance physical and thermodynamic properties of DNA storage sets.**

***Index Terms*— DNA storage, DNA sets design, DBWO, end-constraint.**

I. Introduction

With the wide application of network and the development of various industries, the amount of data is increasing exponentially. By 2025, 463 exabytes (EB) of data are expected to be produced every day around the world, and global data is expected to reach 175 zettabytes (ZB), which is 159 times the number of observable stars [1]. With data growing so rapidly, storing such a huge amount of information becomes an urgent problem. Traditional storage media, such as flash, HDD, and tape, all have disadvantages such as short storage time, high cost, easily lost data, and environmental pollution [2]. Therefore, looking for a new storage medium is particularly necessary.

DNA is a natural storage medium, and it is capable of being a widely-used storage tool due to its high storage density, abundant resources, easy access, long storable time, low energy consumption, and other advantages [3]. With the rapid development of DNA storage technology, the widespread application of DNA storage is being further advanced [4], [5]. In a nutshell, DNA storage includes six steps: coding, synthesis, preservation, retrieval, sequencing, and decoding. In recent years, DNA storage technologies have been constantly developed and applied. In 2012, Church *et al.* [6] proposed a new coding method, using next-generation technologies to encode arbitrary digital information. In 2015, Yazdi *et al.* [7] used a new storage architecture and coded the Wikipedia pages of six universities; the results showed that the architecture had a high storage efficiency. Furthermore, they [8] implemented a portable, error-correctable DNA storage method and encoded data using an integrated processing pipeline. In 2020, Tomek *et al.* [9] extracted a unique file from a database which stored 5 TB of data through chemical processing, thus driving the practical scalability of high-capacity data storage systems. Lee *et al.* [10] reported a multiplexed enzymatic DNA synthesis method using maskless photolithography.

In the decoding process, it is easy to occur errors of deletion, insertion, and replacement [11]. In order to minimize the emergence of these errors and to compress the size of data, scholars have proposed various coding methods to construct DNA storage sets. In 2013, Goldman *et al.* [12] adopted the Huffman coding method in DNA storage for the first time, effectively increasing the coding potential to 1.58 bit/nt. However, the storage density corresponding to this coding method was only 0.33 bit/nt. In order to avoid excessive data redundancy, in 2016, Bornholt *et al.* [13] used exclusive XOR coding to improve Goldman's coding scheme, which has more than twice the coding density of the Huffman coding method. In order to further increase the coding density, degenerate bases were proposed by Hwang and Bang [14] to encode binary data for achieving high information capacity. Blawat *et al.* [15] innovated the forward error correction coding method. Eight bits binary data were stored in five bases according to the coding table, and data density of the method reaches 0.92 bit/nt. In 2017, Erlich and Zielinski [16] put forward the fountain coding method. That study raised the

Manuscript received February 7, 2021; revised April 22, 2021, June 2, 2021, and September 1, 2021; accepted October 13, 2021. Date of publication October 18, 2021; date of current version April 1, 2022. 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, and Grant 61802040; in part by the Liaoning Revitalization Talents Program under Grant XLYC2008017; in part by the Innovation and Entrepreneurship Team of Dalian University under Grant XQN202008; in part by the Natural Science Foundation of Liaoning Province under Grant 2021-MS-344; in part by the Liaoning BaiQianWan Talents Program; and in part by the General Project of the Education Department of Liaoning Province under Grant LJKZ1186. *(Corresponding authors: Bin Wang; Qiang Zhang.)*

The authors 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: juneqiongqiong@gmail.com; zhengyanfen95@gmail.com; wangbinpaper@gmail.com; zhangq@dlut.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNB.2021.3121278, provided by the authors.

Digital Object Identifier 10.1109/TNB.2021.3121278

theoretical limit of the coding potential to an unprecedented value of 1.98 bit/nt, and significantly reduced the redundancy required for error correction. The RS coding approach proposed by Rashtchian *et al.* [17] also increased the possibility of retrieving correct original information under the case of deletion, insertion, and replacement error by adding redundancy. In 2018, Song *et al.* [18] discovered a new method to construct DNA storage sets using the run-length limit constraint and GC-content constraint. Cao *et al.* [19] and Yin *et al.* [20] proposed new coding methods, which effectively improved the lower bound of DNA storage coding and enhanced the quality of coding. Moreover, a new coding method that could correct an error of deletion, insertion, or replacement in the coding process and generate a sequence with 50% GC content was innovated by Xue and Lau [21] in 2020.

When coding for storage, quality as well as quantity is required. In this work, a new meta-heuristic algorithm Double-strategy Black Widow Optimization Algorithm (DBWO) is applied and combined with combinatorial constraints to construct high quality and quantity of sequences in DNA storage. Based on Black Widow Optimization Algorithm, the random switch and double-weight offspring strategies are appended in DBWO. To evaluate if DBWO has strong exploration and exploitation abilities which are two vital features of an algorithm, we use 26 benchmark functions to test its performance. Exploration is to find a promising solution that is not adjacent to the current solution within a certain range, and this ability is associated with preventing local optimal stagnation [22], [23]. Exploitation is a kind of local search in a certain space to find a better solution, which is related to the improvement of convergence speed [24]. Results show that the two strategies improve the shortcomings of slow convergence speed and easily falling into local optima. In addition, DBWO is used to design DNA storage sets, and the lower bounds are improved observably. Although the number of sequences is increased, their properties are less stable. Due to the hybridization character of bases, the End-constraint is utilized to keep the properties of the DNA storage sets in a more stable state. In order to estimate the validity of this constraint, we evaluate physical and thermodynamic properties of sequence sets by the number of hairpin structures, the stability of melting temperature, and minimum free energy. The comparative results show that the End-constraint can effectively enhance the properties of DNA storage sets. The quantity of DNA storage sets is improved by the DBWO algorithm's strong exploration and exploitation abilities, and the qualities of sets are guaranteed by combinatorial constraints.

The organizational structure of this whole work is described as follows: Section 2 describes BWO, the two strategies used in DBWO, and results of 26 benchmark functions in detail; Section 3 introduces the details and results of constructing DNA storage sets; Section 4 introduces the End-constraint and the comparisons of physical and thermodynamic properties in DNA sets under the innovative constraint; and Section 5 is a summary of this work and the prospect of future targets.

II. Algorithm Description

A. Black Widow Optimization Algorithm

In 2019, Hayyolalam and Kazem [25] innovated a meta-heuristic optimization algorithm called Black Widow Optimization Algorithm (BWO) from the behavior of spiders. In this algorithm, the actions of female spiders are divided into generation of offspring, cannibalism, and mutation. The method to generate the offspring is shown in (1), where φ is a random number from 0 to 1, x_1 x_2 and y_1 y_2 represent parent and offspring respectively. Cannibalism refers to the improvement of population diversity by comparing and then replacing the offspring which have bigger fitness. Fitness in algorithm refers to measure the superiority of an individual in the population. In this paper, fitness is mainly calculated through different benchmark functions, and the smaller the fitness is, the better the individual performance is. Mutation is inspired by the Genetic algorithm and refers to a random exchange of values on a certain dimension of two candidate sets in the offspring pool.

$$\begin{aligned} y_1 &= \varphi \times x_1 + (1-\varphi) \times x_2 \\ y_2 &= \varphi \times x_2 + (1-\varphi) \times x_1 \end{aligned} \tag{1}$$

Compared to some other meta-heuristic algorithms, the performance of BWO in exploration and exploitation are somewhat improved. However, the updating method of the whole population is relatively simple, which may result in insufficient exploration. Moreover, the logic of parent generation is simple, which tends to result in inappropriate diversity of the population and converging on local optima. To avoid these shortcomings, we improve BWO so that it can be better applied to solve practical problems.

B. Double-Strategy Black Widow Optimization Algorithm

In order to improve the above shortcomings of BWO, we introduce two strategies: random switch and double-weight offspring [26]. The exploration and exploitation abilities of DBWO are effectively measured through the experimental results of 26 benchmark functions.

1) Random Switch Strategy: The random switch strategy is used to improve the availability of current solutions of BWO. We compare α with β to determine whether to implement this strategy on a global scope or not, as shown in (2) and (3). Cauchy random number α has a small peak value at the origin and a large fluctuation at both ends, so it can be used as an available value to generate fluctuations and jump out of local optima [27].

$$\alpha = \tan(\pi \times (rand - 0.5)) \tag{2}$$

$$\beta = 1 - {CIter}/{MIter} \tag{3}$$

CIter means the number of the current iteration, *MIter* means the maximum iteration number, and *rand* is a random number from 0 to 1. When α is smaller than β, one dimension of the candidate solution will be replaced by a random dimension in the current optimal solution; otherwise, the candidate solution will be preserved. This strategy improves the population diversity and exploitation ability of the algorithm, and will make the current solution approach the optimal value indefinitely [28].

2) Double-Weight Offspring Strategy: In the process of updating offspring, we introduce the double-weight offspring strategy. When weight factors are large, the algorithm will search for candidate solutions far away from the current solution, which can prevent falling into local optima. When weight

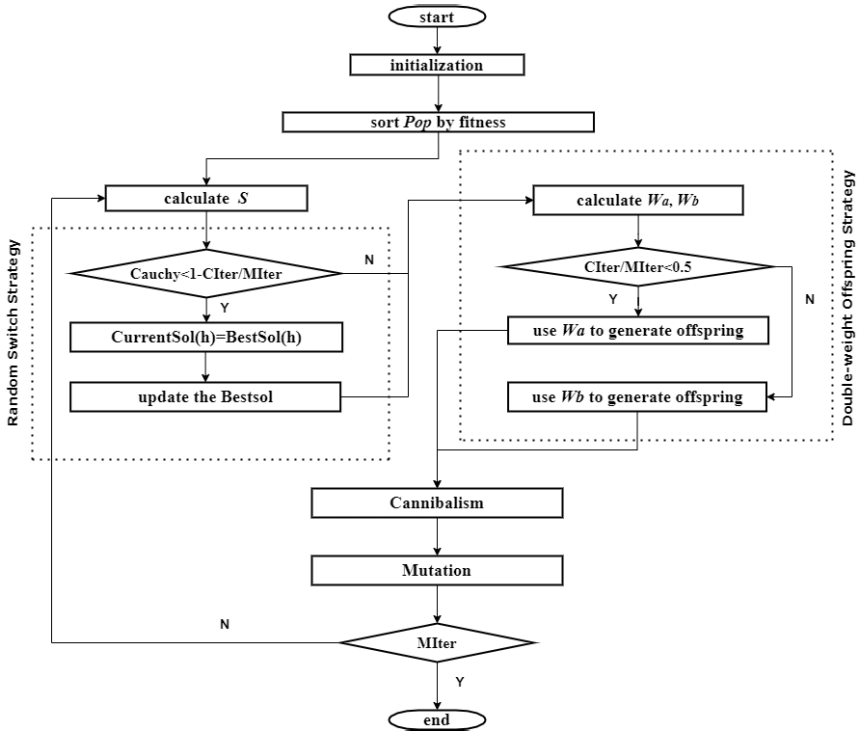

Fig. 1. Flow chart of DBWO.

factors are small, the local search capability of algorithm will be strong, thus enhancing the convergence speed of the algorithm [29]. To implement this property, weight factors W_a and W_b are introduced into BWO.

$$W_a = \left(1 - CIter/MIter\right)^{1-(rand-0.5)\times S/MIter} \quad (4)$$

$$W_b = \left(2 - 2\times CIter/MIter\right)^{1-(rand-0.5)\times S/MIter} \quad (5)$$

$$CIter/MIter < 0.5 \quad (6)$$

The values of weights W_a and W_b are calculated by (4) and (5), and parameter S in them is used to control the update range of the population. It is initialized as 0 and if the result of the current update does not reach an ideal value, S will be divided by two to prevent S from becoming too large. On the contrary, if the population falls into local optima, S will be increased by one to expand the search range.

Equation (6) is used to estimate whether the current iteration is in the first half of the total cycle or not; if it is, offspring will be updated with W_a, if not, offspring will be updated with W_b. The specific selection method is shown by (7) and (8).

$$\begin{cases} y_1 = W_a \times rand. \times x_1 + (1 - rand). \times x_2 \\ y_2 = W_a \times rand. \times x_2 + (1 - rand). \times x_1 \end{cases} \quad CIter/MIter < 0.5 \quad (7)$$

In the first half of the cycle, the parameter W_a is introduced, and (7) is used to update offspring population. The value range of W_a is smaller than that of W_b, which enables the population to finely search around the parent generation, thus enhancing the optimization ability and local search ability of DBWO.

$$\begin{cases} y_1 = rand. \times x_1 + W_b \times (1 - rand). \times x_2 \\ y_2 = rand. \times x_2 + W_b \times (1 - rand). \times x_1 \end{cases} \quad CIter/MIter \geq 0.5 \quad (8)$$

In the second half of the cycle, the parameter W_b is introduced, and (8) is applied to update offspring population. The range of W_b is wide and it fluctuates more with each iteration, and can thus enhance the global random search capability of algorithm. In this way, the generation equations of the offspring are changed from (1) to (7) and (8). Therefore, the whole population focuses on exploration in the first half of the cycle and exploitation in the second half of the cycle, thus improving the convergence and optimization abilities of DBWO [30].

3) Double-Strategy Black Widow Optimization Algorithm: Fig. 1 and Fig. 2 show the flowchart and pseudo-code of DBWO, respectively.

The specific implementation logic of population updating and selection in DBWO are as follows:

Step 1: Initialize the population and sort it according to fitness, then store the result into *Pop*;

Step 2: The parameter S from (4) and (5) is calculated. If the fitness of the present solution is less than the current optimal solution, S is divided by two to control the size. If not, then S is increased by one;

Step 3: Equations (2) and (3) are used to determine whether the random switch strategy is used or not. If α is smaller than β, the random switch strategy is used to replace a dimension in the present solution. If not, it will not be used;

Step 4: Equations (4) and (5) are used to generate parameters W_a and W_b in the double-weight offspring strategy, and (6) is used to determine which parameter to use. W_a is used to generate the offspring in the first half of the algorithm cycle according to (7). W_b is used to generate the offspring in the second half of the algorithm cycle according to (8);

Step 5: Sort the *Pop* and delete the poorly performing candidate set, then store the result into *Pop2*;

TABLE I
RESULT OF BENCHMARK FUNCTIONS (F1—F5)

F		F1	F2	F3	F4	F5
PSO	**Mean**	1.66E−04	2.08E+05	6.49E+03	4.31E−04	7.76E+02
	Median	2.67E−05	1.95E+05	5.48E+03	4.78E−05	5.88E+02
DBWO	**Mean**	**0**	**4.55E+01**	1.06E-12	**0**	**0**
	Median	**0**	**4.55E+01**	5.72E-13	**0**	**0**
BWO	**Mean**	2.40E−04	1.13E+02	**0**	0	2.73E−01
	Median	1.74E−04	1.01E+02	**0**	0	1.53E−01
ABC	**Mean**	2.40E−04	3.18E+02	1.05E+03	3.71E−05	1.60E+01
	Median	1.74E−04	3.09E+02	8.33E+02	2.72E−05	1.02E+01
BBO	**Mean**	1.86E−13	2.19E+03	3.29E+02	1.51E−108	1.04E−03
	Median	1.02E−14	2.11E+03	3.20E+02	0	4.20E−04

Algorithm 1 DBWO Algorithm Pseudo Code

```
//Initialization
Initialize parameters nPop, MIter, nvar, nMutation, pCannibalism
Calculate the best solution so far.
Put the initialized population into Pop
WHILE stop criterion is not satisfied do
//Procreation and Cannibalism
Generate the parameter S of the Double-weight Offspring Strategy
For i=1 to nvar do
  If CIter/MIter<0.5 then
    y1=Wa×rand.×x1+(1-rand).×x2
    y2=Wa×rand.×x2+(1-rand).×x1
  Else
    y1= rand.×x1+Wb×(1-rand).×x2
    y2= rand.×x2+ Wb×(1-rand).×x1
  End if
End for
Calculate the fitness and replace the pCannibalism
Put the population into Pop2
//Random Switch Strategy
For l=1 to nPop do
  For h=1 to nvar
    If tan(π×(rand-0.5))<1-CIter/MIter then
     Pop(l).Position(h)=BestSol.Position(h)
    End if
  Sort the population by fitness
  End for
End for
//Mutation
For k=1 to nMutation
  Randomly swap two elements of a sequence
  Put the population into Pop3
End for
//Update
Pop=Pop+Pop2+Pop3
Sort the population by fitness
Return the best solution BestOfAllDBWO
End while
```

Fig. 2. The pseudo-code of DBWO algorithm.

Step 6: Randomly exchange values on two dimensions for *nMutation* offspring, and store the result into *Pop3*;

Step 7: Add *Pop2* and *Pop3* into *Pop* and return the optimal value.

The algorithm flow chart in Fig. 1 explains the execution process of DBWO, and the pseudo-code in Fig. 2 presents the overall logic of DBWO.

C. Benchmark Function Comparison

This work uses 26 benchmark functions to evaluate the performance of exploration and exploitation in DBWO. For the validity and availability of data results, each experimental result is averaged over 30 runs. In order to better measure the performance, we selected several representative heuristic algorithms to compare with DBWO [31]–[33]. Particle Swarm Optimization Algorithm (PSO) is a longstanding and widely-used algorithm with long development time. Artificial Bee Colony Algorithm (ABC) and Biogeography-Based Optimization Algorithm (BBO) are both recent algorithms, and BWO is the original algorithm on which DBWO is based. The abbreviations and descriptions of algorithms are shown in Table. S1. We compare different hyperparameters in our experiments, setting number of dimensions to 10, 20, and 50; population size to 100, 150, and 200; and iteration number to 500, 1000, and 1500. Table. I and Table. II are test results of benchmark functions F1–F10 when Nvar is 50, Npop is 200 and Maxiter is 1500, and the complete experimental results are shown in Table S2. The parameters' meaning in Table. I and Table. II are as follows, Nvar represents population dimensions, Npop represents the number of species, Maxiter represents the biggest cycles, Best represents the best value in the operation. The standard test function tables are shown in Table. S3. In Table. I and Table. II, Mean and Median represent the average and median fitness of the algorithm after 30 runs, respectively. The best results of corresponding functions are shown in bold.

The exploitation ability of DBWO is measured through unimodal functions (F1–F5). Unimodal test functions do not have local minima, and their search space only has one global minimum. Algorithms with good global exploitation performance can quickly find the optimal solution and not fall into local optima, so this benchmark function is very suitable for evaluating convergence and capability of exploitation in an algorithm [34]. From the test function results of F1, F4 and F5, DBWO can converge to the theoretical optimal value, indicating that its exploitation ability has been greatly improved compared to other algorithm BWO. However, there is still a gap between the global optimal value and the theoretical optimal value of the algorithm in F3, illustrating that the exploitation ability of the algorithm can still be improved.

The overall exploration capability of DBWO is measured by the multimodal optimization benchmark functions (F6–F10). The exploration capability refers to the ability to approach the optimal value indefinitely, and only algorithms with strong exploration ability perform well [35]. When Nvar is 50, Npop is 200, and Maxiter is 1500, global theoretical optimal values were obtained for F6–F10, showing that DBWO has the strongest optimization ability among other algorithms, and its abilities are greatly improved compared with BWO.

In order to show the optimization performance of DBWO more intuitively, this work selects the convergence iteration

TABLE II
RESULT OF BENCHMARK FUNCTIONS (F6—F10)

F		F6	F7	F8	F9	F10
PSO	**Mean**	4.31E−04	7.76E+02	2.79E+01	2.79E+01	2.79E+01
	Median	4.78E−05	5.88E+02	2.64E+01	2.64E+01	2.64E+01
DBWO	**Mean**	**0**	**0**	**0**	**0**	**0**
	Median	**0**	**0**	**0**	**0**	**0**
BWO	**Mean**	0	2.73E−01	7.57E−03	7.57E−03	7.57E−03
	Median	0	1.53E−01	1.07E−03	1.07E−03	1.07E−03
ABC	**Mean**	3.71E−05	1.60E+01	9.07E+00	9.07E+00	9.07E+00
	Median	2.72E−05	1.02E+01	9.30E+00	9.30E+00	9.30E+00
BBO	**Mean**	1.51E−108	1.04E−03	4.90E−01	4.90E−01	4.90E−01
	Median	0	4.20E−04	4.30E−01	4.30E−01	4.30E−01

curves of several benchmark functions. From Fig. S1 and Fig. S2, we can see that DBWO greatly improves the convergence speed and optimization ability compared with other heuristic algorithms.

III. CONSTRUCT DNA STORAGE SETS

The coding problem is NP complete, so we can use meta-heuristic algorithms to solve it. Constraints can help to construct more suitable DNA sequences for storage. Algorithms with strong exploration and exploitation abilities can improve the lower bounds of DNA storage sets, that is, construct a larger number of sequences which can be used in DNA storage. In this section, DBWO with constraints is introduced in designing DNA storage sets and the process is described in detail.

A. Constraints

Constraints are used to filter appropriate sequences that meet combinatorial conditions. To select more stable and suitable DNA sequences for storage, we use combinatorial constraints to design DNA storage sets. Continuous bases could lead to errors in the process of decoding continuous bases and cause unmanageable hybridization reactions of the whole sequence [36], [37], so run-length limit constraint (RLL) is used to limit its occurrence [38]. RLL means the absence of two consecutive identical bases, in other words, the absence of homopolymer. Moreover, due to the different characteristics of GC and AT base pairing, the GC-content constraint [18] is introduced in this work, which means the GC content of a sequence is maintained at 50%. Finally, Hamming distance constraint is used to maintain the stability of similarity between sequences. The constraint set $S^{GC,RLL}(n,d)$ is a storage set representing a sequence of length n that conforms to the GC-content constraint, run-length limit constraint, and Hamming distance of d. The sequences constructed by these combinatorial constraints can effectively avoid insertion, deletion, and replacement errors in the process of sequencing and decoding.

B. Experiment Environment and Symbol

The results of the entire simulation experiment are run on a desktop computer with an Intel Core I7 3.6-GHz processor and 4 GB storage space, and uses MATLAB 2018. The experimental results are shown in Tables IV–IX, and better results of each table are shown in bold. Table. III illustrates the meanings of different superscripts in this work.

TABLE III
THE MEANING OF SUBSCRIPTS

subscript	meaning
M	The result is obtained by DMVO
D	The result is obtained by DBWO
B	The result is obtained by BWO
E	The result contents End-constraint
N	The result does not content End-constraint

In order to represent the bases more conveniently, the letters of each gene are coded by the quaternary numbers, where 0, 1, 2, and 3 represent the bases T, C, G, and A, respectively.

C. The Progress of DNA Storage Coding

DNA storage sets design can be realized by three strategies: the first is search strategy, which selects sequences that meet the constraints from total storage sets; the second is template mapping strategy, where a template set T of binary sequences and a map set M of binary sequences are used to generate a DNA storage set that meets requirements; the third is evolutionary strategy, where a set of DNA sequences is randomly generated, and a DNA storage set conforming to combinatorial constraints is finally produced.

So that all sequences used for storage can realize Watson-Crick hybridization and avoid non-specific hybridization, we combined search and evolutionary strategies and applied DBWO to construct candidate sets of DNA storage sequences that meet combinatorial constraints. The procedures that apply DBWO to construct DNA storage sets are as follows.

Step 1: Initialize parameters and generate the candidate sets that satisfy combinatorial constraints, then put them into *Pop*;

Step 2: Generate the parameter of the double-weight offspring strategy;

Step 3: Calculate the Hamming distances of candidate sets and replace the worse individuals, then put the sets into *Pop2*;

Step 4: Use the random switch strategy to update the candidate sets and sort sets by Hamming distance;

Step 5: Randomly exchange values on two dimensions in the candidate sets to update them and store the result into *Pop3*;

Step 6: Reserve the candidate sets that satisfy the combinatorial constraints.

TABLE IV
DBWO VS. DMVO IN LOWER BOUND OF $S^{GC,RLL}$ (n,d)

n\d	3	4	5	6	7	8	9
4	12^{M}						
	12^{D}						
5	20^{M}	8^{M}					
	20^{D}	**8^{D}**					
6	58^{M}	24^{M}	8^{M}				
	60^{D}	**28^{D}**	**8^{D}**				
7	125^{M}	45^{M}	16^{M}	6^{M}			
	127^{D}	**46^{D}**	**16^{D}**	**6^{D}**			
8	324^{M}	94^{M}	32^{M}	13^{M}	5^{M}		
	326^{D}	**110^{D}**	**37^{D}**	**14^{D}**	**5^{D}**		
9	713^{M}	223^{M}	64^{M}	24^{M}	10^{M}	5^{M}	
	799^{D}	**227^{D}**	**72^{D}**	**27^{D}**	**11^{D}**	**5^{D}**	
10	1906^{M}	555^{M}	151^{M}	54^{M}	19^{M}	9^{M}	4^{M}
	1979^{D}	**584^{D}**	**155^{D}**	**57^{D}**	**22^{D}**	**9^{D}**	**4^{D}**

As mentioned above, we first generate sequence candidate sets content the combinatorial constraints by using random strategy. Then, three behaviors and two strategies of DBWO are used to evolve the initial candidate sets. Finally, the appropriate DNA storage sets are selected.

D. The Improvement of Lower Bound

Table. IV shows the result of DBWO applied to construct DNA storage sets. In order to evaluate the ability of DBWO to improve lower bound of storage coding sets, we compare the results of DBWO and previous work DMVO [19]. From Table. IV, it can be seen that the lower bounds obtained by DBWO are improved by 3%–72% compared with previous work; under combinatorial constraints, superior candidate sets of DNA sequences can be obtained. The longer sequence length and the smaller Hamming distance are, the better DBWO performs. The data in the table shows that DBWO, with its strong exploitation and exploration abilities, can improve the lower bounds of DNA storage sets.

IV. ENHANCEMENT OF DNA SETS PROPERTIES

In DNA storage, data is broken up into small chunks, converted into base sequences, and then stored in DNA sequences. If we have to repeat the PCR amplification for each sequence, it will certainly cause low storage efficiency. The general way to solve this problem is to add some non-payload, such as primer and address bit [39]. the primer bit is added before the data block in order to distinguish different files; address bit is added after primer to mark the order of sequences [13]. In this section a new constraint called End-constraint is introduced, which can enhance the properties of DNA storage sets. To illustrate its validity, we measure the physical property of the sequence in terms of the number of hairpin structures, and the thermodynamic property through melting temperature and minimum free energy. High-quality DNA storage sets can reduce the error rate and are more suitable for DNA storage [12].

A. End-Constraint

When sequences are stored in the double-stranded form, the G-C base pair is connected by three hydrogen bonds, whereas the A-T base pair is connected by two hydrogen bonds. The 3' end is the key to amplification, and if there is a GC mismatch at the 3' end, the strand is difficult to disentangle. In addition, the presence of multiple GC bases in the 3' end has a significant impact on the thermodynamic and physical properties of an entire sequence. Thus, we use End-constraint to limit the GC content in the 3' end. The End-constraint means in a sequence *S (s1, s2, s3…sn)*, the number of G or C in the last five bases cannot exceed three [9]. For example, the sequence TAGTCAGCG has four Gs and Cs in the last five bases, which not content the End-constraint. The End-constraint is expressed in the equation below:

$$\left|G_{S_{Last5}}\right| + \left|C_{S_{Last5}}\right| < 3 \tag{9}$$

where $\left|G_{S_{Last5}}\right|$ and $\left|C_{S_{Last5}}\right|$ represent the number of G and C in the last five bases of sequence S.

In this paper, the RLL constraint, GC-content constraint, Hamming distance constraint, and End-constraint are combined to enhance physical and thermodynamic properties of DNA storage sets. The constraint set $S^{GC,RLL,END}(n,d)$ is a coding set that represents a sequence of length n that conforms to the GC content constraint, run-length limit constraint, and End-constraint, and has the Hamming distance of d.

Table. V shows the number of sequence candidate sets screened by DBWO under the constraint of $S^{GC,RLL,END}(n,d)$. In combination of constraint conditions, smaller Hamming distance constraints causes DBWO to take more advantage of its exploration and exploitation abilities, thus improving the lower bound of DNA storage sets significantly.

In order to better illustrate the quality of candidate sets under the End-constraint, we measure sequences in terms of physical and thermodynamic properties. The physical property is mainly evaluated by the number of hairpin structures, and the thermodynamic property is evaluated by melting temperature and minimum free energy. Table. VI–Table. XIII are comparisons between sets with and without the End-constraint.

TABLE V
DBWO LOWER BOUNDS WITH END-CONSTRAINT

n\d	**3**	**4**	**5**	**6**	**7**	**8**	**9**
5	20	8					
6	32	13	5				
7	106	40	16	6			
8	178	71	24	10	5		
9	527	189	48	23	9	3	
10	1187	376	108	41	12	7	4

TABLE VI
COMPARISON OF HAIRPIN STRUCTURE NUMBER WITH AND WITHOUT END-CONSTRAINT

n\d	3	4	5	6	7	8	9
8	147^{N}	48^{N}	14^{N}	5^{N}	2^{N}		
	74^{E}	**28^{E}**	**9^{E}**	**3^{E}**	**1^{E}**		
9	1134^{N}	327^{N}	103^{N}	42^{N}	14^{N}	9^{N}	
	690^{E}	**267^{E}**	**60^{E}**	**34^{E}**	**10^{E}**	**4^{E}**	
10	5979^{N}	1729^{N}	447^{N}	166^{N}	81^{N}	30^{N}	13^{N}
	3535^{E}	**1070^{E}**	**306^{E}**	**112^{E}**	**38^{E}**	**18^{E}**	**11^{E}**

TABLE VII
COMPARISON OF THE RATIO OF HAIRPIN STRUCTURES WITH AND WITHOUT END-CONSTRAINT

n\d	3	4	5	6	7	8	9
8	0.45^{N}	0.44^{N}	0.38^{N}	0.36^{N}	0.40^{N}		
	0.41^{E}	**0.39^{E}**	**0.37^{E}**	**0.30^{E}**	**0.20^{E}**		
9	1.42^{N}	1.44^{N}	1.43^{N}	1.56^{N}	1.27^{N}	1.80^{N}	
	1.31^{E}	**1.41^{E}**	**1.25^{E}**	**1.48^{E}**	**1.11^{E}**	**1.33^{E}**	
10	3.02^{N}	2.96^{N}	2.88^{N}	2.91^{N}	3.68^{N}	3.33^{N}	3.25^{N}
	2.98^{E}	**2.84^{E}**	**2.83^{E}**	**2.73^{E}**	**3.17^{E}**	**2.57^{E}**	**2.75^{E}**

TABLE VIII
COMPARISON OF $\overline{F_{Tm}(S)}$ WITH AND WITHOUT END-CONSTRAINT

n/d	3	4	5	6	7	8	9
8	5.84^{N}	14.66^{N}	6.20^{N}	6.09^{N}	9.01^{N}		
	3.56^{E}	**2.71^{E}**	**4.67^{E}**	**4.11^{E}**	**1.32^{E}**		
9	4.60^{N}	4.69^{N}	5.17^{N}	4.03^{N}	4.03^{N}	4.23^{N}	
	3.43^{E}	**3.67^{E}**	**2.61^{E}**	**2.83^{E}**	**3.99^{E}**	**2.37^{E}**	
10	4.36^{N}	4.63^{N}	4.36^{N}	5.28^{N}	**3.51^{N}**	4.63^{N}	4.77^{N}
	3.23^{E}	**3.16^{E}**	**3.49^{E}**	**2.44^{E}**	3.65^{E}	**4.05^{E}**	**0.32^{E}**

B. Hairpin Structure

Hairpin structures are a secondary structure of a DNA molecule that consists of two parts: a hairpin diameter and a ring [39]. It is a structure formed by hybridization reactions between two complementary parts in a DNA sequence. Fig. 3 shows a hairpin structure formed by the sequence AACGCATTCGGTGTA.

If a sequence forms a hairpin during the storage procedure, its physical structure is unstable and cannot be stored as a single strand. Hairpins may cause the stored data to be corrupted and unreadable, so it is necessary to avoid sequences that may cause hairpin structures during coding. We use (10) to calculate the possible number of hairpin structures that the sequence $S(s_1,s_2,s_3,\ldots,s_n)$ can form.

$$f_{Hairpin}(S) = \sum_{r}^{(n-2pinlen)} \sum_{k=pinlen+[r/2]}^{(n-pinlen-[r/2])} Hairpin(S,k) \quad (10)$$

where r and *pinlen* represent the shortest subsequence length needed to form a hairpin ring and a hairpin stem, respectively. In a sequence, if a hairpin structure is produced at the k^{th} base of the sequence, and if the number of complementary bases in the stem of a hairpin is more than half of the total stem length, *Hairpin(S,k)* is 1, otherwise, it is 0.

Table. VI shows the comparison of total numbers of hairpin structures in sequence candidate sets screened by DBWO under $S^{GC,RLL,END}(n,d)$ and $S^{GC,RLL}(n,d)$ constraints. These sequences correspond to the data marked with subscript D in Table. IV and Table. V. The sequence results with superscript E are obtained under $S^{GC,RLL,END}(n,d)$, and results with superscript N are obtained under $S^{GC,RLL}(n,d)$. According to the data in Table. VI, when n is larger and d is smaller, the difference between the two constraints is more obvious. Moreover, the number of hairpin structures formed by the sequence candidate sets with End-constraint is 38% to 98% lower than without the constraint; in every case it can always obtain better results, which indicates that the End-constraint can enhance physical property of sequence candidate sets.

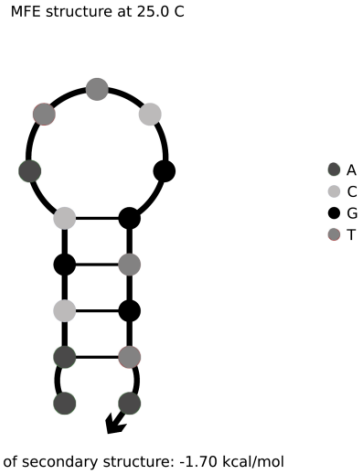

Fig. 3. Hairpin structure formed by sequence AACGCATTCGGTGTA.

Table. VII is the comparison of the ratios of hairpin structures in sequences screened by $S^{GC,RLL,END}(n,d)$ and $S^{GC,RLL}(n,d)$. The smaller ratio is, the more stable the physical property of a sequence is. From the data in Table. VII, it can be seen that the sequences under the End-constraint have smaller ratios in every case, indicating that the number of corresponding hairpin structures is greatly reduced. The ratios of $S^{GC,RLL,END}(8,d)$ is reduced by 7%–50%, $S^{GC,RLL,END}(9,d)$ is reduced by 2%–25%, and $S^{GC,RLL,END}(10,d)$ is reduced by 1%–23%. The dramatic reduction of hairpin structures shows that the candidate set of sequences under the End-constraint has more stable physical property.

C. Melting Temperature

The stability of melting temperature (Tm) of whole storage sets can be judged by comparing the Tm of each sequence with the mean value of Tm [40]. In the process of sequencing, especially annealing of the PCR amplification reaction, all need to speculate the Tm values are important. When storage sets are used for DNA storage, if the differences in Tm between sequences are large, it is difficult to choose an appropriate annealing temperature. Therefore, in order to maintain the stability of sequences, Tm values of sequences in DNA storage sets are generally required to be consistent, which effectively reduces the probability of generating incomplete matching double strands [41].

For the candidate set $S(s_1, s_2, s_3 \ldots s_m)$ with m sequences, we use $F_{Tm}(S)$ to represent the difference in Tm values between sequences. The calculation equation is as follows:

$$F_{Tm}(S) = \sum_{i=1}^{m} \left\{ Tm(S_i) - \overline{Tm}(S) \right\}^2 \quad (11)$$

where $Tm(S_i)$ represents the Tm of the i^{th} sequence in candidate set S, and $\overline{Tm}(S)$ represents the mean Tm of S.

Table. VIII shows the comparison of $\overline{F_{Tm}(S)}$ screened by DBWO under constraints $S^{GC,RLL}(n,d)$ and $S^{GC,RLL,END}(n,d)$. $\overline{F_{Tm}(S)}$ represents the stability of Tm in the whole set and it is the ratio of $F_{Tm}(S)$ under corresponding constraint to sequence set number. The larger

TABLE IX
COMPARISON OF $\delta(G)$ WITH AND WITHOUT END-CONSTRAINT

n/d	3	4	5	6	7	8	9
8	0.40^{N}	0.41^{N}	0.41^{N}	0.42^{N}	0.50^{N}		
	$\mathbf{0.29^{E}}$	$\mathbf{0.35^{E}}$	$\mathbf{0.29^{E}}$	$\mathbf{0.17^{E}}$	$\mathbf{0.07^{E}}$		
9	0.42^{N}	0.39^{N}	0.43^{N}	0.49^{N}	0.37^{N}	0.29^{N}	
	$\mathbf{0.36^{E}}$	$\mathbf{0.38^{E}}$	$\mathbf{0.36^{E}}$	$\mathbf{0.37^{E}}$	$\mathbf{0.24^{E}}$	$\mathbf{0.28^{E}}$	
10	0.43^{N}	0.44^{N}	0.43^{N}	0.47^{N}	$\mathbf{0.26^{N}}$	$\mathbf{0.43^{N}}$	0.51^{N}
	$\mathbf{0.36^{E}}$	$\mathbf{0.35^{E}}$	$\mathbf{0.37^{E}}$	$\mathbf{0.25^{E}}$	0.34^{E}	0.54^{E}	$\mathbf{0.06^{E}}$

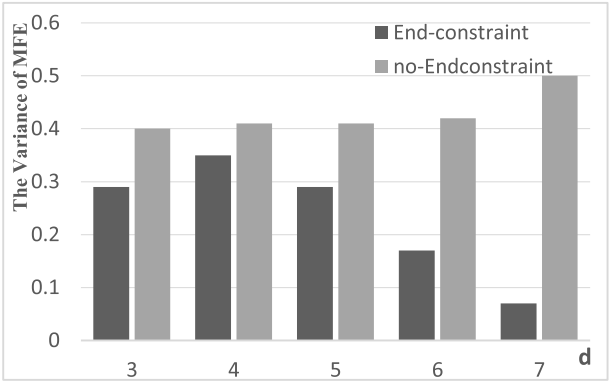

Fig. 4. The comparison between $\delta(G)$ of $S^{GC,RLL,END}(\delta d)$ and $S^{GC,RLL}(8,d)$.

TABLE X
COMPARISON OF THERMODYNAMIC PROPERTY WITH AND WITHOUT END-CONSTRAINT

TYPE	n=20\d	16	17	18	19
END	$\overline{F_{Tm}(S)}$	**0.90**	**0.87**	**0.27**	**0.02**
	$\delta(G)$	**0.40**	**0.22**	**0.04**	**0.07**
NO-END	$\overline{F_{Tm}(S)}$	1.53	1.54	1.89	1.03
	$\delta(G)$	1.08	0.25	0.60	0.72

TABLE XI
COMPARISON OF THERMODYNAMIC PROPERTY WITH AND WITHOUT END-CONSTRAINT

TYPE	n=30\d	25	26	27	28
END	$\overline{F_{Tm}(S)}$	**0.21**	**0.19**	**0.09**	0.52
	$\delta(G)$	**0.26**	**0.09**	1.00	**0.01**
NO-END	$\overline{F_{Tm}(S)}$	1.21	0.69	0.25	**0.26**
	$\delta(G)$	2.14	0.55	**0.28**	0.12

TABLE XII
COMPARISON OF HAIRPIN STRUCTURE IN $S^{GC,RLL}(20,d)$ WITH AND WITHOUT END-CONSTRAINT

n=20\d	16	17	18	19
End	**737**	**673**	**617**	**290**
No-End	900	670	618	448

$\overline{F_{Tm}(S)}$ is, the greater the difference in Tm values of sequences in S is. According to the table, in $\overline{F_{Tm}(S)}$, $S^{GC,RLL,END}(8,d)$ is 24%–85% lower than $S^{GC,RLL}(8,d)$, $S^{GC,RLL,END}(9,d)$ is 21%–49% lower than $S^{GC,RLL}(9,d)$, $S^{GC,RLL,END}(10,d)$ is 13%–93% lower than $S^{GC,RLL}(10,d)$, and $S^{GC,RLL,END}(n,d)$ can obtain smaller $\overline{F_{Tm}(S)}$ under almost all conditions in the table. All results indicate that the End-constraint can design sets with more stable Tm values.

D. Minimum Free Energy

The minimum free energy (MFE) of a sequence is the minimum value among free energies of all possible results [42]. Minimum free energy is an important thermodynamic parameter to measure hybridization reaction, it refers to the energy released by hybridization reaction. The more energy released by two sequences in a hybridization reaction means the structure between them is more stable, so we can measure the stability of a structure through the magnitude of minimum free energy. The minimum free energy in this work is calculated using PairFold [43].

Table. IX shows the comparison of $\delta(G)$ between candidate sequence sets screened by DBWO under combinatorial constraints $S^{GC,RLL,END}(n,d)$ and $S^{GC,RLL}(n,d)$. $\delta(G)$ is the variance of minimum free energy, represent the stability of MFE in the whole set. According to Table. IX, $\delta(G)$ of $S^{GC,RLL,END}(8,d)$ is 15%–86% lower than that of $S^{GC,RLL}(8,d)$, $S^{GC,RLL,END}(9,d)$ is 2%–35% lower than $S^{GC,RLL}(9,d)$, and $S^{GC,RLL,END}(10,d)$ is 14%–88% lower than $S^{GC,RLL}(10,d)$. The sequences with the End-constraint can obtain smaller $\delta(G)$ in nearly all cases in the table, indicating that the sequence candidate sets under the End-constraint have more stable MFE.

Fig. 4 shows the comparison of $\delta(G)$ between storage sets under $^{SGC,RLL}(8,d)$ and $S^{GC,RLL,END}(8,d)$. In every case, MFE all has insecure state intuitively illustrates that sequences without End-constraint have poor manifestation. However, sequences under the End-constraint have better performance in $\delta(G)$, especially when $d=7$. The melting temperature and minimum free energy are important aspects of thermodynamic property [40], so in this paper we use them to measure the thermodynamic property of sequences. Through Tables VII-XIII and Fig. 4, we can see the DNA storage sets under the End-constraint have more consistent Tm and minimum free energy, which indicate that thermodynamic properties of sequences under the new constraint are observably improved.

In addition to filtrating out short sequences with stable physical and thermodynamic properties, End-constraint can also works well for longer sequence. Table. X and Table. XI show the comparison of $\overline{F_{Tm}(S)}$ and $\delta(G)$ in the conditions of $S^{GC,RLL}(n,d)$ and $S^{GC,RLL,END}(n,d)$ with $n = 20$ and $n = 30$, respectively. In the table, END represents the sequences under the End-constraint, and NO-END represents the sequences that not under the End-constraint. From the data in the table, it can be seen that under almost all conditions, End-constraint can be used to select the sequence with more stable melting temperature and minimum free energy. This is because fewer GCs at the end ensures the sequence be more easily untangled during splitting operation, thus ensuring a small difference in the temperature of the unwinding chain compared with other sequences.

By comparing the number of hairpin structures under $S^{GC,RLL}(n,d)$ and $S^{GC,RLL,END}(n,d)$ with $n = 20$ and $n = 30$, the results can be seen from the data in the Table. XII and Table. XIII that sequences with End-constraint are less likely to have non-specific hybridization in the hybridization process, which indicates that their physical property is more stable.

TABLE XIII
COMPARISON OF HAIRPIN STRUCTURE IN $S^{GC,RLL}(30,d)$ WITH AND WITHOUT END-CONSTRAINT

n=30\d	25	26	27	28
End	**2962**	**2023**	**1342**	**1669**
No-end	4029	4274	2788	2596

In addition to being used as primer bit and address bit, longer sequences can also be used to data bit. Sequences with more stable physical and thermodynamic properties are not easy to break or bring hybridization reaction in the storage process, and the melting temperature is more stable in the process of sequencing. Using such sequences to store data can ensure the safety of data storage.

Through the evaluation of hairpin structures, melting temperature, and minimum free energy, it is clear that DNA storage sets under the End-constraint have more stable physical and thermodynamic properties. Sequences with less hairpin structure can effectively avoid non-specific hybridization, whereas stable melting temperature and minimum free energy enable more stable storage in double-stranded DNA. These sequences in DNA storage can improve the quality of sequences by minimizing non-specific hybridization and enabling data to be stored in DNA sequences more stably. In addition to the sequence candidate sets with stable physical and thermodynamic properties have a large role in realizing DNA storage, other possible methods [44]–[53] also have great promise.

V. CONCLUSION

When coding for logical redundancy, sequences with more differences can avoid the occurrence of non-specific hybridization as far as possible, whereas sequences with relatively stable physical and thermodynamic properties are less likely to create problems in the storage procedure. In this paper, we propose the Double-strategy Black Widow Optimization Algorithm and apply it in DNA storage with combinatorial constraints. The algorithm improves optimization performance mainly through two strategies: random switch and double-weight offspring strategies. In order to intuitively demonstrate the improved performance of DBWO, we compare the results of DBWO and four other meta-heuristic algorithms with 26 different test functions. The results show that DBWO obtains the best value in almost every test function and every condition, achieving the theoretical optimal value in 15 test functions, whereas improving results by 1–10 orders of magnitude in the other test functions. It performs well especially in the hybrid and composite functions, which shows that the algorithm combines exploration and exploitation abilities well, being able to jump out from local optima to find the global optimum. Further combining DBWO with constraints in constructing DNA sets, it is found that DBWO can enhance the lower bound for data storage by up to 72%. DBWO, with its strong exploitation and exploration abilities can construct high quantities of DNA storage sets.

DNA storage requires not only large quantities but also high quality of sequences. Sequences under the GC-content constraint, run-length limit constraint, and Hamming distance constraint alone have poor performance as measured by physical and thermodynamic properties, therefore, the End-constraint is proposed for designing high-quality sequences based on the hybridization characteristic of bases. To test the effectiveness of the constraint, the properties of constructed sequence sets are measured by counting the number of potential hairpin structures and the stabilities of melting temperature and minimum free energy. DNA sets under the End-constraint have 38%–98% fewer hairpin structures than sets without the constraint, 13%–93% lower variance in melting temperature. In addition, variance of minimum free energy with the constraint is 2%–88% lower than without the novel constraint. A series of experiments illustrate that the physical and thermodynamic properties of sequences under the End-constraint can be enhanced and maintained in a relatively stable state. Sequences under the End-constraint can reduce non-specific hybridization and non-specific amplification.

In future work, we will continue to work on solving coding problems to increase the efficiency and accuracy of DNA storage. In designing DNA storage sets, many constraints related to various properties of sequence are possible, but measuring the effects of a combination of constraints is an unsolved problem. For example, some sequences may exhibit poor melting temperature properties, yet may perform better at contenting other constraints, such as minimum free energy, Hamming distance, and GC-content. Can we include such sequences that perform poorly in only one constraint in candidate sequence sets? In order to better deal with this problem, we plan to assign weight coefficients to constraints based on the contribution rate of each constraint for different scenarios, thereby creating suitable storage sequences for various applications.

REFERENCES

[1] Y. Dong, F. Sun, Z. Ping, Q. Ouyang, and L. Qian, "DNA storage: Research landscape and future prospects," *Nat. Sci. Rev.*, vol. 7, no. 6, pp. 1092–1107, Jun. 2020, doi: 10.1093/nsr/nwaa007.

[2] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using DNA," *Nature Rev. Genet.*, vol. 20, no. 8, pp. 456–466, Aug. 2019, doi: 10.1038/s41576-019-0125-3.

[3] Z. Ping *et al.*, "Carbon-based archiving: Current progress and future prospects of DNA-based data storage," *GigaScience*, vol. 8, no. 6, p. giz075, Jun. 2019, doi: 10.1093/gigascience/giz075.

[4] E. Zhu, F. Jiang, C. Liu, and J. Xu, "Partition independent set and reduction-based approach for partition coloring problem," *IEEE Trans. Cybern.*, early access, Oct. 27, 2020, doi: 10.1109/TCYB.2020.3025819.

[5] 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 Trans. Nanobiosci.*, vol. 17, no. 4, pp. 474–484, Oct. 2018, doi: 10.1109/TNB.2018.2873221.

[6] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, Sep. 2012, doi: 10.1126/science.1226355.

[7] S. M. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Sci. Rep.*, vol. 5, no. 1, p. 10, Sep. 2015, doi: 10.1038/srep14138.

[8] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 1, pp. 1–6, Jul. 2017, doi: 10.1038/s41598-017-05188-1.

[9] 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, doi: 10.1021/acssynbio.9b00100.

[10] H. Lee *et al.*, "Photon-directed multiplexed enzymatic DNA synthesis for molecular digital data storage," *Nature Commun.*, vol. 11, no. 1, pp. 1–9, Oct. 2020, doi: 10.1038/s41467-020-18681-5.

[11] D. Panda, K. A. Molla, M. J. Baig, A. Swain, D. Behera, and M. Dash, "DNA as a digital information storage device: Hope or hype?" *3 Biotech*, vol. 8, no. 5, p. 239, May 2018, doi: 10.1007/s13205-018-1246-7.

[12] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, pp. 77–80, Jan. 2013, doi: 10.1038/nature11875.

[13] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," *ACM SIGPLAN Notices*, vol. 51, no. 4, pp. 637–649, Jun. 2016, doi: 10.1145/2954679.2872397.

[14] B. Hwang and D. Bang, "Toward a new paradigm of DNA writing using a massively parallel sequencing platform and degenerate oligonucleotide," *Sci. Rep.*, vol. 6, no. 1, pp. 1–7, Nov. 2016, doi: 10.1038/srep37176.

[15] M. Blawat *et al.*, "Forward error correction for DNA data storage," *Proc. Comput. Sci.*, vol. 80, pp. 1011–1022, Jan. 2016, doi: 10.1016/j.procs.2016.05.398.

[16] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–953, Mar. 2017, doi: 10.1126/science.aaj2038.

[17] K. C. Rashtchian *et al.*, "Clustering billions of reads for DNA data storage," in *Proc. Adv. Neural Inf. Process.*, 2017, pp. 3362–3373.

[18] 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, doi: 10.1109/LCOMM.2018.2866566.

[19] 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.

[20] 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," *Int. J. Mol. Sci.*, vol. 21, no. 6, p. 2191, Mar. 2020, doi: 10.3390/ijms21062191.

[21] T. Xue and F. C. M. Lau, "Notice of violation of IEEE publication principles: Construction of GC-balanced DNA with deletion/insertion/mutation error correction for DNA storage system," *IEEE Access*, vol. 8, pp. 140972–140980, 2020, doi: 10.1109/ACCESS.2020.3012688.

[22] D. Castillo-Barnes, F. J. Martinez-Murcia, J. Ramírez, J. M. Górriz, and D. Salas-Gonzalez, "Expectation–maximization algorithm for finite mixture of α-stable distributions," *Neurocomputing*, vol. 413, pp. 210–216, Nov. 2020, doi: 10.1016/j.neucom.2020.06.114.

[23] J. Liu, Z. Zhang, F. Chen, S. Liu, and L. Zhu, "A novel hybrid immune clonal selection algorithm for the constrained corridor allocation problem," *J. Intell. Manuf.*, vol. 1, pp. 1–20, Nov. 2020, doi: 10.1007/s10845-020-01693-9.

[24] X.-L. Zhao, H. Zhang, T.-X. Jiang, M. K. Ng, and X.-J. Zhang, "Fast algorithm with theoretical guarantees for constrained low-tubal-rank tensor recovery in hyperspectral images denoising," *Neurocomputing*, vol. 413, pp. 397–409, Nov. 2020, doi: 10.1016/j.neucom.2020.07.022.

[25] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103249, doi: 10.1016/j.engappai.2019.103249.

[26] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, "An efficient double adaptive random spare reinforced whale optimization algorithm," *Expert Syst. Appl.*, vol. 154, Sep. 2020, Art. no. 113018, doi: 10.1016/j.eswa.2019.113018.

[27] E. Candes, L. Demanet, and L. Ying, "A fast butterfly algorithm for the computation of Fourier integral operators," *Multiscale Model. Simul.*, vol. 7, no. 4, pp. 1727–1750, 2009, doi: 10.1137/080734339.

[28] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103300, doi: 10.1016/j.engappai.2019.103300.

[29] M.-A. Ahmadi, M. H. Ahmadi, M. F. Alavi, M. R. Nazemzadegan, R. Ghasempour, and S. Shamshirband, "Determination of thermal conductivity ratio of CuO/ethylene glycol nanofluid by connectionist approach," *J. Taiwan Inst. Chem. Engineers*, vol. 91, pp. 383–395, Oct. 2018, doi: 10.1016/j.jtice.2018.06.003.

[30] M. G. Kostenko, A. I. Gusev, and A. V. Lukoyanov, "Disorder–order and order–order phase transformations in Ta_5C_4 phases predicted using the evolutionary algorithm and symmetry analysis," *Phys. Chem. Chem. Phys.*, vol. 22, no. 41, pp. 24116–24132, Oct. 2020, doi: 10.1039/d0cp03842c.

[31] Y. Wang, J. Lv, L. Zhu, and Y. Ma, "Crystal structure prediction via particle-swarm optimization," *Phys. Rev. B, Condens. Matter*, vol. 82, Sep. 2010, Art. no. 094116, doi: 10.1103/PhysRevB.82.094116.

[32] D. L. Wagner, D. R. Novog, and R. R. LaPierre, "Genetic algorithm optimization of core-shell nanowire betavoltaic generators," *Nanotechnology*, vol. 31, no. 45, Nov. 2020, Art. no. 455403, doi: 10.1088/1361-6528/aba86d.

[33] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007, doi: 10.1007/s10898-007-9149-x.

[34] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113338, doi: 10.1016/j.eswa.2020.113338.

[35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377, doi: 10.1016/j.eswa.2020.113377.

[36] 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, doi: 10.1109/TNB.2020.2971644.

[37] J. Zhang, R. Chiodini, A. Badr, and G. Zhang, "The impact of next-generation sequencing on genomics," *J. Genet. Genomics*, vol. 38, no. 3, pp. 95–109, Mar. 2011, doi: 10.1016/j.jgg.2011.02.003.

[38] 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, doi: 10.1109/LCOMM.2019.2912572.

[39] P. K. Zuber *et al.*, "The universally-conserved transcription factor RfaH is recruited to a hairpin structure of the non-template DNA strand," *eLife*, vol. 7, p. e36349, May 2018, doi: 10.7554/eLife.36349.

[40] M. R. Shortreed *et al.*, "A thermodynamic approach to designing structure-free combinatorial DNA word sets," *Nucleic Acids Res.*, vol. 33, no. 15, pp. 4965–4977, Sep. 2005, doi: 10.1093/nar/gki812.

[41] J. Sager and D. Stefanovic, "Designing nucleotide sequences for computation: A survey of constraints," *DNA Comput.*, vol. 3892, pp. 275–289, Jun. 2006.

[42] S. Kawashimo *et al.*, "Dynamic neighborhood searches for thermodynamically designing DNA sequence," in *DNA Computing*. Berlin, Germany: Springer, 2008, pp. 130–139.

[43] 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, doi: 10.1016/j.jmb.2004.10.082.

[44] L. Organick *et al.*, "Random access in large-scale DNA data storage," *Nature Biotechnol.*, vol. 36, no. 3, p. 242, Mar. 2018, doi: 10.1038/nbt.4079.

[45] L. Organick *et al.*, "Probing the physical limits of reliable DNA data retrieval," *Nature Commun.*, vol. 11, no. 1, pp. 1–7, Jan. 2020, doi: 10.1038/s41467-020-14319-8.

[46] S. K. Tabatabaei *et al.*, "DNA punch cards for storing data on native DNA sequences via enzymatic nicking," *Nature Commun.*, vol. 11, no. 1, pp. 1–10, Apr. 2020, doi: 10.1038/s41467-020-15588-z.

[47] 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, Jan. 2020, doi: 10.1016/j.csbj.2020.08.004.

[48] T. Song, S. Pang, S. Hao, A. Rodríguez-Patón, and P. Zheng, "A parallel image skeletonizing method using spiking neural P systems with weights," *Neural Process. Lett.*, vol. 50, no. 12, pp. 1485–1502, Oct. 2019, doi: 10.1007/s11063-018-9947-9.

[49] Y. Zheng, J. Wu, and B. Wang, "CLGBO: An algorithm for constructing highly robust coding sets for DNA storage," *Frontiers Genet.*, vol. 12, May 2021, doi: 10.3389/fgene.2021.644945.

[50] F. Yang, Q. Zou, and B. Gao, "GutBalance: A server for the human gut microbiome-based disease prediction and biomarker discovery with compositionality addressed," *Briefings Bioinf.*, vol. 22, no. 5, p. bbaa436, Jan. 2021, doi: 10.1093/bib/bbaa436.

[51] B. Cao, X. Zhang, J. Wu, B. Wang, Q. Zhang, and X. Wei, "Minimum free energy coding for DNA storage," *IEEE Trans. Nanobiosci.*, vol. 20, no. 2, pp. 212–222, Apr. 2021, doi: 10.1109/TNB.2021.3056351.

[52] S. Zhou, P. He, and N. Kasabov, "A dynamic DNA color image encryption method based on SHA-512," *Entropy*, vol. 22, no. 10, p. 1091, Sep. 2020, doi: 10.3390/e22101091.

[53] A. Witte, Á. Muñoz-López, M. Metz, M. R. Schweiger, P. Janning, and D. Summerer, "Encoded, click-reactive DNA-binding domains for programmable capture of specific chromatin segments," *Chem. Sci.*, vol. 11, no. 46, pp. 12506–12511, Dec. 2020, doi: 10.1039/d0sc02707c.