Mathematical Biosciences and Engineering

MBE, 19(12): 14142–14172.
DOI: 10.3934/mbe.2022659
Received: 16 August 2022
Revised: 06 September 2022
Accepted: 12 September 2022
Published: 26 September 2022

http://www.aimspress.com/journal/MBE

Research article

An enhanced whale optimization algorithm for DNA storage encoding

Sijie Wang[1,*], Shihua Zhou[1] and Weiqi Yan[2]

[1] Key laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, China

[2] School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand

* **Correspondence:** Email: wangsijie@s.dlu.edu.cn; Tel: +8641187402106; Fax: +8641187403733.

Abstract: Metaheuristic algorithms have the drawback that local optimal solutions are prone to precocious convergence. In order to overcome the disadvantages of the whale optimization algorithm, we propose an improved selective opposition whale optimization algorithm (ISOWOA) in this paper. Firstly, the enhanced quasi-opposition learning (EQOBL) is applied to selectively update the position of the predator, calculate the fitness of the population before and after, and retain optimal individuals as the food source position; Secondly, an improved time-varying update strategy for inertia weight predator position is proposed, and the position update of the food source is completed by this strategy. The performance of the algorithm is analyzed by 23 benchmark functions of CEC 2005 and 15 benchmark functions of CEC 2015 in various dimensions. The superior results are further shown by Wilcoxon's rank sum test and Friedman's nonparametric rank test. Finally, its applicability is demonstrated through applications to the field of biological computing. In this paper, our aim is to achieve access to DNA files and designs high-quantity DNA code sets by ISOWOA. The experimental results show that the lower bounds of the multi-constraint storage coding sets implemented in this paper equals or surpasses that of previous optimal constructions. The data show that the amount of the DNA storage cods filtered by ISOWOA increased 2–18%, which demonstrates the algorithm's reliability in practical optimization tasks.

Keywords: whale optimization algorithm; enhanced quasi-opposition learning; time-varying inertia weight; DNA storage set

1. Introduction

Many metaheuristic algorithms [1–3] have been derived from research of the collective behavior of social organisms. These are generally employed to solve global optimization problems. They simulate nature and human intelligence to achieve the optimal solution. There are three main categories [4–9], as shown in Figure 1.

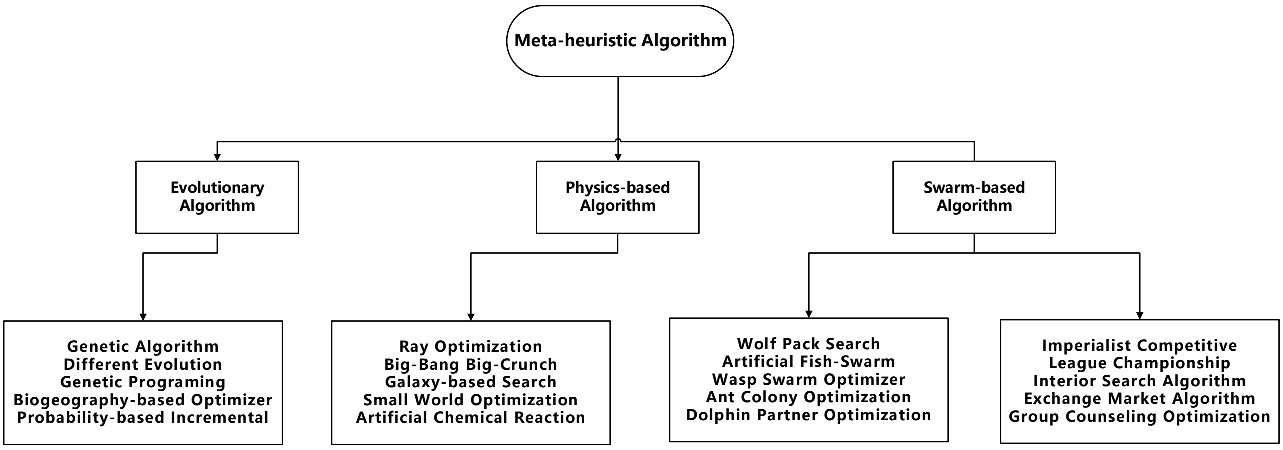

Figure 1. Classification of metaheuristic algorithms.

The first is based on the evolutionary process. For example, one of the first population-based stochastic algorithms [2], the genetic algorithm (GA), imitates the theory of evolution, with steps of dynamic adaptive adjustment to gain the most suitable chromosome. Similar algorithms include differential evolution (DE), which relies on mutation strategies and control parameters and whose values are difficult to determine. Zhou et al. [1] developed a self-adaptive differential evolution algorithm to greatly address a single batch-processing machine scheduling problem. Deng et al. [2] proposed a DE algorithm with wavelet basis function (WMSDE), which integrated a wavelet basis function and normal distribution, and improved the local search information and diversity of solutions. Zhao et al. [3] proposed a two-stage cooperative evolutionary algorithm with problem-specific knowledge to address energy-efficient scheduling of no wait flow shop problem (EENWFSP). Most evolutionary algorithms imitate biological characteristics, a series of mechanisms, such as competition orcooperation, are required to complete evolution.

In the real world, biological individuals lack complicated intelligence, but based on certain rules, biological groups experience interaction, the interaction between the individual and the environment can often produce evolutionary phenomena. This is the basis of swarm intelligence optimization. By observing biological populations, we are able to simulate their cooperative strategies, such as hunting, swarming, migration, and foraging, and apply them to system design and optimization. Kaveh et al. [4] presented the water strider algorithm (WSA), inspired by the lifestyle of water strider bugs, to tackle discrete structural design problems, such as health monitoring. Zhao et al. [5] proposed a cooperative water wave algorithm (CWWO) to address the DANIFSP with the goal of minimizing the maximum assembly completion time. Yapici et al. [6] proposed a novel algorithm that imitates the leadership hierarchy of swarms, which is called the pathfinder algorithm (PFA). We thus review the crowd-based metaheuristic intelligent optimization algorithms, such as poor and rich optimization (PRO) [7] and gaining sharing knowledge (GSK) [8], the latter was motivated by human lifespan. It involves junior and senior gaining, both followed by a sharing phase.

Algorithms based on physical rules and chemical reactions usually simulate physical laws in nature, such as gravity, annealing, collision, and inertial force, and apply them to combinatorial constraint optimization. Liu et al. [9] proposed a metaheuristic photon search algorithm (PSA) with good search efficiency and convergence speed. Pflug et al. [10] employed a new random gradient metaheuristic algorithm for structural optimization problems.

Metaheuristic algorithm is based on heuristic algorithm. It is a combination of random algorithm and local search algorithm. Through constantly iterative optimization, the exploration and exploitation of search space can be realized. From the practical perspective, these kinds of algorithms rarely require the characteristics of functions in the optimization field, and have strong robustness to the optimization of functions. As metaheuristic algorithm has the advantages of a simple principle, easy programmability, and few parameters, a slew of novel metaheuristic algorithms have emerged to solve NP problems in different fields. Recently, the algorithm has been applied to continuous constraint problems and discrete structural design problems in the fields of computer science, engineering optimization design [11], optimal scheduling [12], resource coordination and scheduling [13], multi-objective optimization [14] and combinatorial optimization [15]. Deng et al. [2] applied DE to an assignment problem of airport gate which was a large-scale NP problem with complex constraint. Dhiman et al. [16] illustrated MoSSE by combing the characteristics of spotted hyena optimizer (SHO) with other metaheuristics. After testing on IEEE CEC-9 standard test function, the findings showed the stability of MOSSE to deal with the real complex multi-objective optimization problems. Zhao et al. [17] proposed a cooperative multi-stage hyper-heuristic (CMS-HH) algorithm to address six specific combinatorial optimization problems. The experimental results demonstrate the efficiency and significance of CMS-HH algorithm. Yang et al. [18] proposed an improved DBSCAN optimized by arithmetic optimization algorithm (AOA) for achieving automatic optimization of parameters. The experimental data show that DBSCAN can provide highly accurately clusters more efficiently.

Whale optimization algorithm (WOA) [19] was proposed by Mirjalili, which mainly realized the optimization by simulating the behaviors of humpback whale. Through the test of 29 benchmark functions and the application of six practical engineering problems, the algorithm was proved that its better accuracy and speed of convergence than other metaheuristic algorithms. Nevertheless, we see from the literature that WOA is constantly changing the position of the leader, but it still leads to prematurity when the leader is trapped in the fake optimal solution.

In view of the shortcomings of WOA, a large number of scholars have conducted research on WOA. Many of them improved whale optimization algorithm through different strategies and mechanisms. Chakraborty & Saha et al. [20] integrated the modified mutual mechanism from symbiotic organisms search with WOA to alleviate inherent drawback of premature convergence. Jin et al. [21] proposed an algorithm with random evolution and special reinforcement called REWOA. Luo & Chen et al. [22] proffered a multi-strategy boosted mutative whale optimization algorithm (MBMWOA), which was integrated with the chaotic initial regulation and the "shrinking" strategy of local search. Yang et al. [23] proposed a cellular automaton mechanism to improve whale optimization algorithm for the robust extreme learning machine (ELM) training. The experiment showed the improved optimizer was around 7% superior to the basic.

In addition, in related work, the algorithm was improved by combining with other novel metaheuristic algorithms. For example, Fan & Chen et al. [24] proposed to combine WOA with the characters of salp swarm algorithm (SSA) and learning mechanism. Nguyen et al. [25] fused galactic swarm optimization (GSO) with the variant of the WOA to avoid early convergence. In addition, WOA is also broadly used in practical applications. Due to the simplicity and accuracy of ISP problems, Abdel-Basset et al. [26] presented a hybrid metaheuristic algorithm called HSMA_WOA. The performance of HSMA_WOA had been proved by chest radiographs.

In summary, WOA has strong adaptability to continuous optimization problems in various application fields. However, it has fewer applications in discretization applications. Zhang & Hong et al. [27] proposed a WOA integrated the dynamically adjusted weight and improved search strategy to solve the traveling salesman problem library (TSPLIB). Wang et al. [28] put forward an improved algorithm called CMWOA that contained chaotic and other improved strategies to solve difficult diagnosis problems. Hemmelmayr et al. [29] harnessed an adaptive large neighborhood search algorithm (ALNS) with whale optimization algorithm to get higher quality solutions in a multi-compartment capacitated arc routing problem. To sum up, WOA algorithm has strong global search ability in discretization, so it also has advantages in building DNA storage code sets. In view of the characteristics of high density and high persistence of DNA, DNA storage has broken through the difficulties of traditional computers in data storage and computing speed, and has become a feasible solution. Then for the DNA storage problem, the first consideration is the DNA coding problem. An effective model that stores a large amount of data in a small amount of DNA is crucial to improve the storage rate, that is, the model can construct a larger number of coding sets within a certain base length. In order to solve this key problem, it is necessary to construct enough coding sets under the same sequence length, further improve the lower bound of the coding set, so that it can store more information at a given length, and thus improve the storage coding rate. This is also the meaning of DNA storage coding research.

In this paper, a combinatorial strategy is applied to optimize the WOA to achieve better convergence performance, the optimized algorithm is applied to address the very large-scale discrete problems in the field of biological computing. The major contributions have been outlined as follows：

1) In this paper, we propose the enhanced quasi-opposite learning (EQOBL). During each iteration, the locations of individual whales are selectively updated to better expand the search scope of the whale and enhance WOA's ability to explore the space. By continuously reducing the distance difference between whale (the candidate individual) and prey (the global optimal individual), the convergence speed is accelerated, and the main objective of improving the quality and diversity of the whale population is finally achieved.

2) An improved nonlinear decreasing strategy of inertia weight with collaborative control parameter is proposed to improve the convergence accuracy of WOA and adjust the global and local search ability to find the optimum.

3) Considered the application in the construction of DNA storage code sets, by discretizing the whale optimization algorithm and using its special search rules, it can achieve breakthrough in the lower bounds under the same coding constraints and improve coding rate. The increased coding rate means that ISOWOA can effectively reduce the synthesis cost of sequences under the same performance requirements. Based on the characteristics of fourth-generation nanopore sequencing technology, the DNA storage codes after screening by ISOWOA can minimize the errors in sequencing and complete the access to large DNA files.

The rest of this paper is briefly summarized as follows: In the second section, we briefly represent the concept of WOA; In the third section, we propose the new combination strategy to enhance the

WOA; In the fourth section, we demonstrate the results of the CEC standard test function experimental simulation, convergence image analysis, multi-dimensional test analysis and statistical discussion; In the fifth section, we showcase its advantages in building DNA storage coding set in the field of biological computing, and discuss the experimental results in detail. Finally, the research work is summarized and the future work is envisioned.

2. Whale optimization algorithm

Humpback whales are generally 13–15 m, which are well known for their complex calls, extra-long front wings, and leaps out of the water. Lacking teeth and being unable to hunt large prey like other whales, humpbacks have evolved a different way of hunting—diving into deep water, creating bubble nets, and spiraling upward around their prey. Inspired by bionics, the whale optimization algorithm [19] was devised. The process is modeled as the three phases [19].

2.1. Shrinking of the prey encirclement

Prey can be located and encircled. The position of individual represents a candidate solution. If the location of the target prey is unknown, WOA will assume that the optimal location of the current whale is the target prey's location. Candidate solutions will continuously approach to the optimal solution in subsequent iterations. Through comparative analysis of location information, the optimal candidate solution will be updated using its positions. Humpback whales constantly shrink the encirclement till the target prey is found. The update formula [19] is

$$\overrightarrow{Dis} = \left|\overrightarrow{C\_Factor} \cdot \overrightarrow{Whale^*}(t) - \overrightarrow{Whale}(t)\right| \tag{1}$$

$$\overrightarrow{C\_Factor} = 2 \cdot \vec{r} \tag{2}$$

$$\overrightarrow{Whale}(t+1) = \overrightarrow{Whale^*}(t) - \overrightarrow{A\_Factor} \cdot \overrightarrow{Dis} \tag{3}$$

$$\overrightarrow{A\_Factor} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{4}$$

where $\overrightarrow{Dis}$ is the distance from the optimal solution to the current individual whale in the jth-dimensional position. If $\overrightarrow{Whale}(t+1)$ has a better positional advantage than $\overrightarrow{Whale^*}(t)$, the optimal position is updated.

2.2. The spiral hunting of humpback whales

The following formula shows the spiral position updating model of WOA, which is denoted [19] as

$$\overrightarrow{Whale}(t+1) = \overrightarrow{Whale^*}(t) + \overrightarrow{Dis^l} \cdot e^{bl} \cdot cos(2\pi l) \tag{5}$$

where · denotes multiplication. To align the probabilities of exploitation and exploration of WOA, the shrinkage encircling mechanism is assumed to have the same probability as the spiral updating position, i.e., $\boldsymbol{P} = \mathbf{0.5}$.

2.3. Random hunting of humpback whales

This strategy updates the location of the search agent based on a randomly selected individual. It is shown mathematically as

$$\begin{cases} \overrightarrow{D\imath s} = \left|\overrightarrow{C\_Factor}\overrightarrow{Whale_{rand}} - \overrightarrow{Whale}\right| \\ \overrightarrow{Whale}(t+1) = \overrightarrow{Whale_{rand}} - \overrightarrow{A\_Factor} \cdot \overrightarrow{D\imath s}, \end{cases} \tag{6}$$

where $\overrightarrow{Whale_{rand}}$ is a random whale, and $\overrightarrow{D\imath s}$ is the distance from the whale to the current individual, and $\overrightarrow{Whale}(t+1)$ is an updated population, and the value of $\overrightarrow{A\_Factor}$ is given by (4).

3. ISOWOA algorithm

Although the whale optimization algorithm has the advantages of a simple principle, easy programmability, and few parameters, it has the drawbacks that local optimal solutions are prone to precocious convergence. The optimization results for various test functions are not ideal. This paper makes the following improvements.

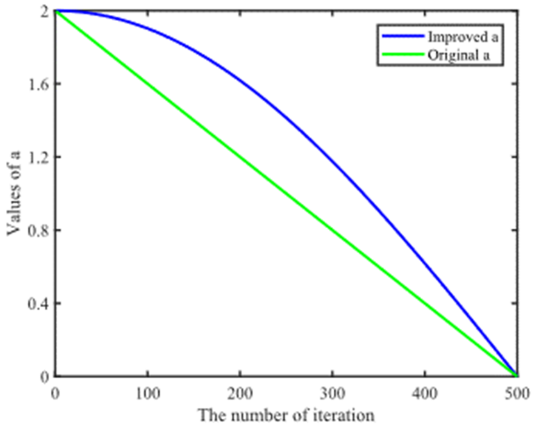

Figure 2. Iterative curve of nonlinear control parameter a.

3.1. Nonlinear control parameter strategy

WOA determines the global or local search through the value of A, which is determined by the factor a. A larger value of a confers a stronger global search ability, and a smaller value confers a stronger local search ability. In the original literature [19], a is linearly decreasing. Hence, the algorithm has a better global exploration ability at the beginning, but it may exist the possibility of easily falling into a local optimum in the middle and late iterations. A nonlinear decreasing parameter is applied to enhance this performance. The model is shown in Figure 2, which keeps a relatively high value for a period of time at the initial phase, then rapidly reduces to a low level, and maintains a low value at the middle and later phase of the iteration. This is expressed as

$$a = 2 * cos\left(\left(\frac{pi}{2}\right) * \left(\frac{t}{Max_{iter}}\right) * log^{n}\right). \tag{7}$$

The nonlinear control parameter a has a high value, which improves the search ability of the ISOWOA. It decays rapidly to a lower value in middle iterations to realize the rapid transition from global search to local search, which is guaranteed through a low value in later iterations.

3.2. Nonlinear decreasing strategy of inertia weight with collaborative convergence factor

The proposed time-varying inertia weight is based on PSO. The parameter ω is important in PSO and plays a decisive role in its overall performance [30]. The focus on optimization algorithms research is to enable the algorithm to find the corresponding global optimal solution on various test functions. Of course, a relatively optimal solution is acceptable in practice. We introduce a time-varying inertia weight ω to improve the convergence accuracy and convergence speed of WOA. It can make the algorithm adaptively adjust the update strategy in order to obtain the global optimal solution. In early iterations, the inertia weight of synergy a makes ISOWOA have stronger global search ability, that is, ISOWOA explores the search space much widely. ω is positively correlated with a and is given by

$$\omega(t) = \left[cos(\frac{pi}{2} * (\omega_{max} - t * (\omega_{max} - \omega_{min})/Max_{iter}))\right] * \frac{a}{(a_{max} - a_{min}) * 0.75}. \tag{8}$$

where a_{max} and a_{min} are the maximum and minimum of the convergence factor; ω_{max} and ω_{min} are defined as 0.8 and 0.4 respectively; and $\mathrm{Max_{iter}}$ defines as 500.

With the novel inertia weight, the three original search methods of WOA are changed to

$$\overrightarrow{Whale}(t+1) \begin{cases} \begin{cases} \omega \cdot \overrightarrow{Whale_{random}}(t) - A \cdot \left|\vec{C} \cdot \overrightarrow{Whale_{random}}(t) - \overrightarrow{Whale}(t)\right| & A \geq 1 \\ \omega \cdot \overrightarrow{Whale^{*}}(t) - cos(2\pi l) \cdot \big(\big|\vec{C} \cdot \big|\overrightarrow{Whale^{*}}(t) - \overrightarrow{Whale}(t)\big) & A < 1 \end{cases} & p < 0.5 \\ \omega \cdot \overrightarrow{Whale^{*}}(t) + \omega \cdot \vec{D} \cdot e^{bl} \cdot cos(2\pi l) & p \geq 0.5 \end{cases} \tag{9}$$

where $b = randi(500), l = (a_2 - 1) * rand + 1$, and $a_2 = -1 - \frac{t}{Max_{iter}}$.

ω is generally controlled by a, decreases nonlinearly, and has stronger adaptability, which is an important factor in balancing the abilities of the ISOWOA. After the coefficient amplification of a, the convergence speed is greatly improved. To avoid wasting computing resources through excessive development, the value of inertial weight decreases significantly with the rapid decline of the nonlinear convergence factor in the middle and later stages of iteration, which greatly enhances local search ability, improves convergence accuracy, avoids precocious convergence, and balances exploitation and exploration.

3.3. Enhanced quasi-opposition learning method (EQOBL)

After Tizhoosh et al. [31] put forward the concept of opposition-based learning, it has been successfully applied in various metaheuristic algorithms. Dhargupta et al. [32] applied it to grey wolf optimization with remarkable results. The strategy has been broadly used in various metaheuristic algorithms. The opposition-based learning strategy can be briefly defined as follows:

$x = (x^1, x^2, \cdots, x^n)$ is a point, where $x^i \in [l^i, u^i]$. $x' = (x^{1'}, x^{2'}, \cdots, x^{n'})$ is defined by its components: $x^{i'} = l^i + u^i - x^i$ [31].

Aiming at the situation that the individual fitness is still too high after opposition learning, in this paper, we propose a variant of OBL, enhanced quasi-opposite learning method (EQOBL) to increase the quality and diversity of candidate solutions, which makes EQOBL participate in each iteration instead of completing the initialization. By continuously reducing the distance difference between whale (the candidate individual) and prey (the global optimal individual), the convergence speed is accelerated. The main objective of improving the quality and diversity of the whale population is finally achieved.

The proceed is described as follows:

1) Define a time-varying factor for judging the difference between the rest and the optimal individual.

2) Implement the EQOBL mechanism for individuals with greater differences than time-varying factors.

3) Among them, the correlation helps to select opposing individuals and opposing dimensions, thus limiting and guiding our search.

4) Then the transformed population was merged with the original population, and the optimal n individuals were selected by the elite mechanism to participate in the subsequent iterative optimization.

Algorithm 1 shows the EQOBL pseudocode, and Figure 3 shows the ISOWOA flowchart.

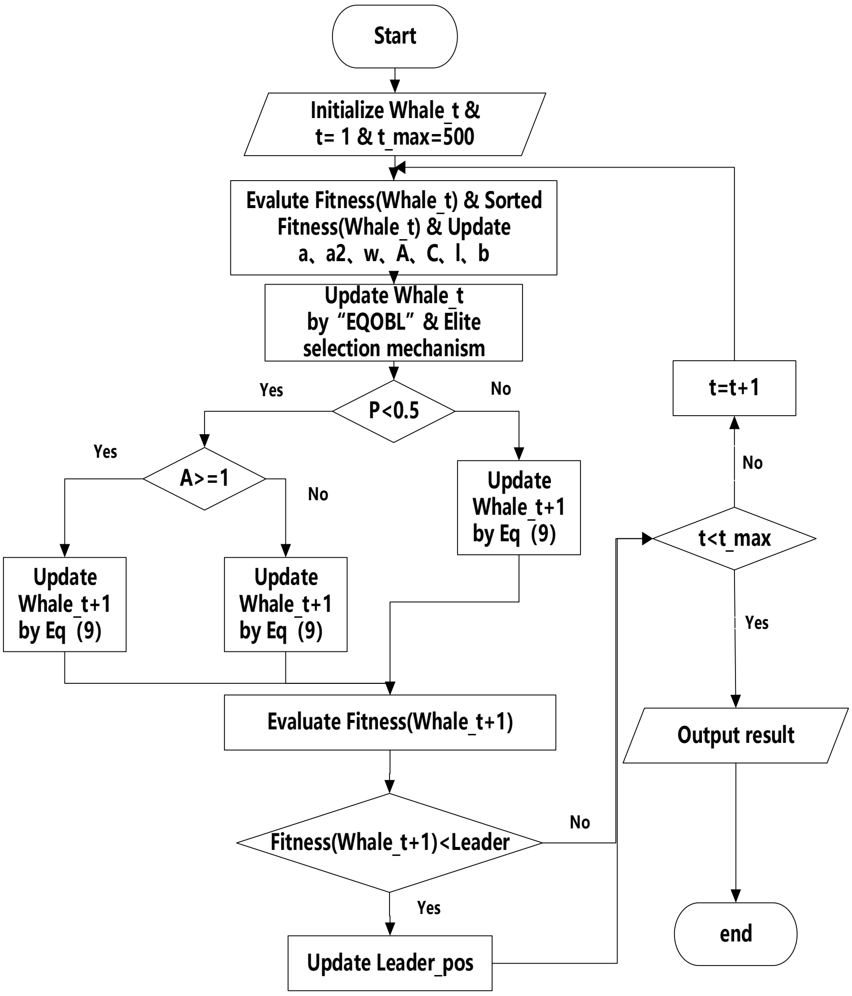

Figure 3. Flowchart of ISOWOA.

Algorithm 1 The pseudocode of EQOBL-based population optimization

```
  Sorted the fitness values of search agents
      $D = 10 * \cos\left(\left(\frac{pi}{2}\right) * \left(\frac{t}{Max_{iter}}\right)\right)$
differece = D, M = (aw + bw) /2, R = rand();
For q = 1 to the count of rest search agents
    For w = 1 to the count of dim
            discrepancy(w) = |Whale(w) - Whale*(w)|
                    If discrepancy (w) > difference
                       number = number + 1
                    End if
     End for
  $correlation = 1 - \frac{6*\sum_{w}(\text{discrepancy }(w))^{2}}{num*(num^{2}-1)}$
If $correlation$ <0
If (dim – number < number)
     else
       For discrepancy (w) > difference
            Whale(w) = R * (u(w) + l(w)) - Whale(w)
            If (Whale(w) < M)
                Whale(w) = M + (Whale(w) - M) * R
            else
                Whale(w) = Whale(w) + (M -Whale(w)) * R
            End if
         End for
    End if
  End if
End for
Update the locations of search agents
Compare the fitness between the two populations
Select the best n individuals as iterative population
```

This strategy can well guide the evolution of the population, its strong convergence enables ISOWOA to more quickly approximate the global optimal value.

4. Experimental simulation results

To better interpret the algorithm's performance, we selected 23 international standard test functions [33]. These 23 functions are composed of three types: 7 unimodal functions (F1–F7), 6 multimodal functions (F8–F13), and 10 fixed-dimension functions (F14–F23). The three types of functions will be compared with the table as the boundary. Unimodal functions enable the investigation of local convergence accuracy and global convergence speed and stability; Multimodal functions can well distinguish global search ability; Fixed-dimension functions can be used to analyze exploration ability and avoidance of local traps. Hence, these test functions can greatly prove the performance of the algorithm. Its mean, standard deviation, iterative convergence curve, and multi-dimensional function tests were compared to those of the basic WOA, several variants, and a modern metaheuristic algorithm. However, in accordance with the No Free Lunch Theorem (NFL) [34], algorithm cannot reach the theoretical extremum on all test functions, and none can achieve the optimal design for all applications. Therefore, algorithms may show superior performance on different functions.

4.1. Effectiveness analysis of three improvement strategies

It can be seen from the above that this paper has made three improvements to the basic WOA algorithm, namely, the nonlinear control parameter strategy, the nonlinear decreasing strategy of inertia weight with collaborative convergence factor and the enhanced quasi-opposition learning method (EQOBL). In order to analyze the effectiveness of the three improved strategies, this section compares the original WOA algorithm with the WOA algorithm that only uses the nonlinear control parameter strategy (named ISOWOA-1), the WOA algorithm that only adds the nonlinear decreasing strategy of inertia weight with collaborative convergence factor (named ISOWOA-2), and the WOA algorithm that only introduces the enhanced quasi-opposition learning method (EQOBL) (named ISOWOA-3), which effectively proves the contribution of each strategy to the algorithm. Here, the widely used CEC 2005 benchmark test function is used as the evaluation standard. The Tables 1 and 2 shows the test results of the four algorithms against the test standard. Black font indicates the best result of the comparison algorithm.

Table 1. Comparison of results of four algorithms on test function set (F1–F7).

Function	Criteria	WOA	ISOWOA-1	ISOWOA-2	ISOWOA-3
F1	ave	5.11E-72	3.6997e-136	**0**	4.2577e-43
	std	2.78E-71	2.0264e-135	**0**	2.2995e-42
F2	ave	2.83E-50	3.6884e-113	**0**	5.1235e-34
	std	1.02E-49	2.0202e-112	**0**	2.8061e-33
F3	ave	5.39E-17	101883.8355	**0**	111093.2992
	std	2.31E-16	30468.7316	**0**	32907.188
F4	ave	47.8759	49.279	**1.1351e-316**	41.824
	std	28.3435	28.0352	**0**	28.721
F5	ave	27.9079	0.52646	**0.0097328**	0.52516
	std	0.37657	1.2255	**0.018956**	0.71357
F6	ave	0.46161	0.14811	**0.0028755**	0.067588
	std	0.29464	0.09739	**0.0042634**	0.042058
F7	ave	0.0029988	0.0040491	**0.00011165**	0.0046454
	std	0.0027323	0.0044947	**0.00010964**	0.0045346

From the experimental results, it can be seen that the improvement of WOA algorithm performance by using nonlinear control parameter strategy alone is limited, but the advantage of this strategy is that it can greatly improve the performance of the algorithm when mixed with other strategies. In addition, from F1–F13, it can be seen that the performance of WOA algorithm is improved most obviously by adding the nonlinear decreasing strategy of inertia weight with collaborative convergence factor alone. The order of magnitude of the convergence value of the function has been greatly improved, which indicates that the nonlinear decreasing strategy of inertia weight with collaborative convergence factor has the largest impact on the optimization performance of WOA algorithm among the three strategies, and further indicates that the nonlinear decreasing strategy of inertia weight with collaborative convergence factor is more effective. It can be seen from the composite functions such as F14 and F20–F23 that the convergence value of the function has been greatly improved, which proves that the introduction of the enhanced quasi-opposition learning method (EQOBL) has significantly improved the performance of the WOA algorithm.

Table 2. Comparison of results of four algorithms on test function set (F8–F14 and F20–F23).

Function	Criteria	WOA	ISOWOA-1	ISOWOA-2	ISOWOA-3
F8	ave	-10232.47	-12550.1708	**-12567.6137**	-12480.7619
	std	1683.6928	88.3732	**4.7451**	376.6853
F9	ave	**0**	**0**	**0**	**0**
	std	**0**	**0**	**0**	**0**
F10	ave	4.59E-15	1.0066e-15	**8.8818e-16**	2.1908e-15
	std	3.11E-15	6.4863e-16	**0**	2.3756e-15
F11	ave	0.0065099	0	**0**	0.032237
	std	0.035656	0	**0**	0.17657
F12	ave	0.022638	0.010272	**8.6621e-05**	0.0041772
	std	0.014756	0.011963	**0.00013368**	0.0024753
F13	ave	0.49925	0.14171	**0.00063103**	0.062659
	std	0.25791	0.16562	**0.00080778**	0.055856
F14	ave	3.2936	2.4049	1.2346	**1.0317**
	std	2.9982	3.0218	0.49925	**0.18364**
F20	ave	-3.1859	-3.1127	-2.963	**-3.2268**
	std	0.18322	0.34223	0.24093	**0.10416**
F21	ave	-8.9355	-9.933	-10.0871	**-10.1283**
	std	2.1799	0.31542	0.085086	**0.050903**
F22	ave	-6.8106	-9.1764	-10.2583	**-10.3815**
	std	3.3228	2.74	0.32565	**0.047403**
F23	ave	-7.0653	-9.8009	-10.406	**-10.5139**
	std	3.3858	1.602	0.13583	**0.048373**

In order to prove the optimization effect of the ISOWOA, we compared it to enhanced whale optimization algorithm (EWOA) [33], improved whale optimization algorithm with joint search mechanism (JSWOA) [35], classical WOA, augmented grey wolf optimizer (AGWO) [33], and adaptive whale optimization algorithm (AWOA) [35]. Among them, EWOA modified the stages of the WOA. JSWOA introduced the joint search mechanism. The data of AGWO and AWOA were from EWOA [33] and JSWOA [35], respectively. The algorithm was tested with the parameters of the comparative algorithms, i.e., a population size of 30, with 500 maximum iterations.

Due to the randomness of relevant parameters, each test function was run 30 times independently. Optimized algorithm performance is usually reflected by the average value and iterative convergence curve. The stability of the optimization algorithm is reflected by the standard deviation. Therefore, the average value and standard deviation of various functions were calculated, as shown in Tables 3–11. Convergence curve is shown in Figures 4–6. Modeling and testing were done with MATLAB R2017a on an AMD Ryzen 7 PC with 16GB RAM and Windows 10. The best mean and variance results are highlighted in boldface in the tables.

4.2. Test of high-dimensional unimodal functions

A unimodal function has only one optimum, which is both local and global, i.e., the enhancement of global search ability helps to search for the global optimum.

To minimize the impact of contingency on the solution of the function, Table 3 lists the evaluation indexes of six metaheuristic algorithms after repeated iteration for 30 times in the unimodal function.

Table 3. Statistical results (average and standard deviation) for unimodal functions.

Function	Criteria	ISOWOA	EWOA	JSWOA	WOA	AGWO	AWOA
F1	ave	**0**	0	0	5.11E-72	6.69E 44	0
	std	**0**	0	0	2.78E-71	1.33E-43	0
F2	ave	**0**	1.40E-195	0	2.83E-50	8.35E-27	6.32E−182
	std	**0**	0	0	1.02E-49	9.41E-27	0
F3	ave	**0**	0	0	5.39E-17	2.98E-08	0
	std	**0**	0	0	2.31E-16	1.08E-07	0
F4	ave	**0**	3.16E-201	**0**	47.8759	1.25E-10	9.31E−179
	std	**0**	0	0	28.3435	5.70E-10	0
F5	ave	**2.11E-02**	2.17E-01	2.79E+01	27.9079	26.9654	2.83E+01
	std	**4.47E-02**	2.81E-01	4.50E−01	0.37657	0.6973	3.36E−01
F6	ave	**4.70E-03**	2.38E-02	7.86E−01	0.46161	1.4381	4.66E−01
	std	**6.50E-03**	3.36E-02	2.20E−01	0.29464	0.3314	2.28E−01
F7	ave	**7.60E-05**	1.78E-04	7.63E-05	0.0029988	1.42E-03	7.96E−05
	std	**6.61E-05**	1.73E-04	7.12E−05	0.0027323	8.21E-04	7.82E−05

In Table 3, ISOWOA has good adaptability and stability when dealing with unimodal functions (F1–F7). Among them, F1–F3 functions all reach the theoretical extreme value in both mean value and standard deviation. In the solutions of F4–F7, for modified variants such as EWOA and JSWOA, the enhanced algorithm improves by one or three orders of magnitude in both mean and standard deviation, which shows that ISOWOA has good convergence accuracy and stability. In addition, F5 is a typical nonconvex function. As the bottom of this function is flat, which is called a valley or banana function. It's a great tool for evaluating the merits of an algorithm. On this function, the optimization value and standard deviation obtained in this paper are the better compared with all other algorithms, which proves that the algorithm has higher optimization accuracy under the premise of ensuring stability.

4.3. Test of high-dimensional multimodal functions

Because multimodal functions have multiple local optimal values, the optimization algorithm needs a stronger global search ability. From the test results of F8–F13 in Table 4, we see that F8, F9, and F11 found the global optimal solution. They converged to the theoretical extreme value. The test results of F12 and F13 were one to four orders of magnitude higher than those of the enhanced metaheuristic algorithm and the variants of WOA, which further confirms the improvements of the proposed strategy. The quality of the sequence set in the iterative process was well controlled.

Table 4. Results of multimodal functions (average and standard deviation).

Function	Criteria	ISOWOA	EWOA	JSWOA	WOA	AGWO	AWOA
F8	ave	**-1.26E+04**	-12569.45	-1.25E+04	-10232.478	-3633.36	−1.23E+04
	std	2.51E-01	**1.63E-01**	1.93E+02	1683.6928	442.5815	6.16E+02
F9	ave	**0**	**0**	0.00E+00	**0**	0.91397	0.00E+00
	std	**0**	**0**	0.00E+00	**0**	5.006	0.00E+00
F10	ave	**8.88E-16**	**8.88E-16**	**8.88E-16**	4.59E-15	9.30E-15	**8.88E-16**
	std	**0**	**0**	0.00E+00	3.11E-15	2.55E-15	0.00E+00
F11	ave	**0**	**0**	0.00E+00	0.0065099	1.18E-03	0.00E+00
	std	**0**	**0**	0.00E+00	0.035656	0.0045	0.00E+00
F12	ave	**9.54E-05**	7.71E-04	4.40E−02	0.022638	0.1024	1.94E−02
	std	**1.07E-04**	2.20E-04	1.52E−02	0.014756	0.0312	1.08E−02
F13	ave	**7.50E-04**	1.44E-02	5.41E−01	0.49925	1.1287	2.68E−01
	std	**9.13E-04**	2.49E-02	1.40E−01	0.25791	0.2193	1.16E−01

Table 5. Results of fixed dimensional multimodal functions (average and standard deviation).

Function	Criteria	ISOWOA	EWOA	JSWOA	WOA	AGWO	AWOA
F14	ave	**9.98E-01**	1.2627	7.01E+00	3.2936	2.3104	2.18E+00
	std	**8.34E-07**	0.68538	5.19E+00	2.9982	2.4804	2.59E+00
F15	ave	**4.03E-04**	9.58E-04	4.96E−04	0.00082536	1.84E-03	**4.03E−04**
	std	**7.04E-05**	7.81E-04	1.84E−04	0.00059597	0.005	7.31E−05
F16	ave	-1.01E+00	**-1.0316**	-1.02E+00	**-1.0316**	**-1.0316**	-1.02E+00
	std	7.08E-03	**1.15E-12**	1.09E−02	1.78E-09	4.17E-06	1.14E−02
F17	ave	4.01E−01	**0.3979**	3.98E−01	**0.3979**	0.3981	3.99E−01
	std	3.39E-03	**8.24E-07**	7.02E−04	2.81E-05	1.39E-05	9.31E−04
F18	ave	3.04E+00	7.5034	**3.00E+00**	3.0001	**3.00E+00**	3.01E+00
	std	5.30E-03	4.9322	5.02E−03	0.00015949	**3.07E-05**	3.58E−02
F19	ave	-3.85E+00	-3.7529	-3.85E+00	-3.8518	**-3.859**	-3.84E+00
	std	**1.63E-03**	0.15693	1.63E−02	0.019943	0.0033	4.71E−02
F20	ave	**-3.19E+00**	-3.0353	-3.14E+00	-3.1859	-3.1858	-3.07E+00
	std	**7.15E-02**	0.23885	1.19E−01	0.18322	0.1199	2.30E−01
F21	ave	**-10.1419**	-10.0167	-1.01E+01	-8.9355	-6.7613	-4.80E+00
	std	**0.02359**	0.2213	**6.07E−03**	2.1799	1.7626	2.04E−01
F22	ave	**-1.04E+01**	-10.228	**-1.04E+01**	-6.8106	-7.1128	-4.98E+00
	std	**0.0040159**	0.27626	5.83E−03	3.3228	1.9801	4.21E−01
F23	ave	**-10.5074**	-10.3471	-1.05E+01	-7.0653	-8.1292	-5.45E+00
	std	**3.30E-03**	0.11708	3.64E−03	3.3858	1.0174	1.57E+00

4.4. Test of high-dimensional fixed-dimension multimodal functions

In Table 5, ISOWOA shows good performance on fixed-dimension multimodal functions. Among the 10 functions tested, there are a total of 6 functions whose average value and standard deviation are all equaled or exceeded those of the other algorithms, they are F14, F15 and F20–F23 functions respectively.

To sum up, 19 of the 23 functions of the algorithm proposed in this paper exceeded or equaled to other algorithms. Among them, F1–F3, F8, F9, and F11 converged directly to the theoretical optimum. Other test results were very close to the theoretical extreme values, achieving the desired effect. In addition, F5 also converged to the theoretical value near 0, which is the best convergence effect for the valley function among the six algorithms. So the proposed algorithm (ISOWOA) maintained a certain stability in many aspects and had good convergence accuracy.

According to the analysis of the experimental data, we see that ISOWOA is the best among the six algorithms in terms of optimization effect and algorithm stability. This further verifies the feasibility of various strategies introduced to address the shortcomings of the algorithm in this paper.

4.5. Convergence analysis

Due to the limited data reduction of the algorithm, only WOA [19], improved whale optimization (IWOA), SCA [36], GWO [37], and MFO [38] were selected as reference objects to draw a function convergence curve. According to the curves in Figures 4–6, we can see that the convergence speed and precision of ISOWOA are better on the 23 test functions. This is because ISOWOA enhances search abilities.

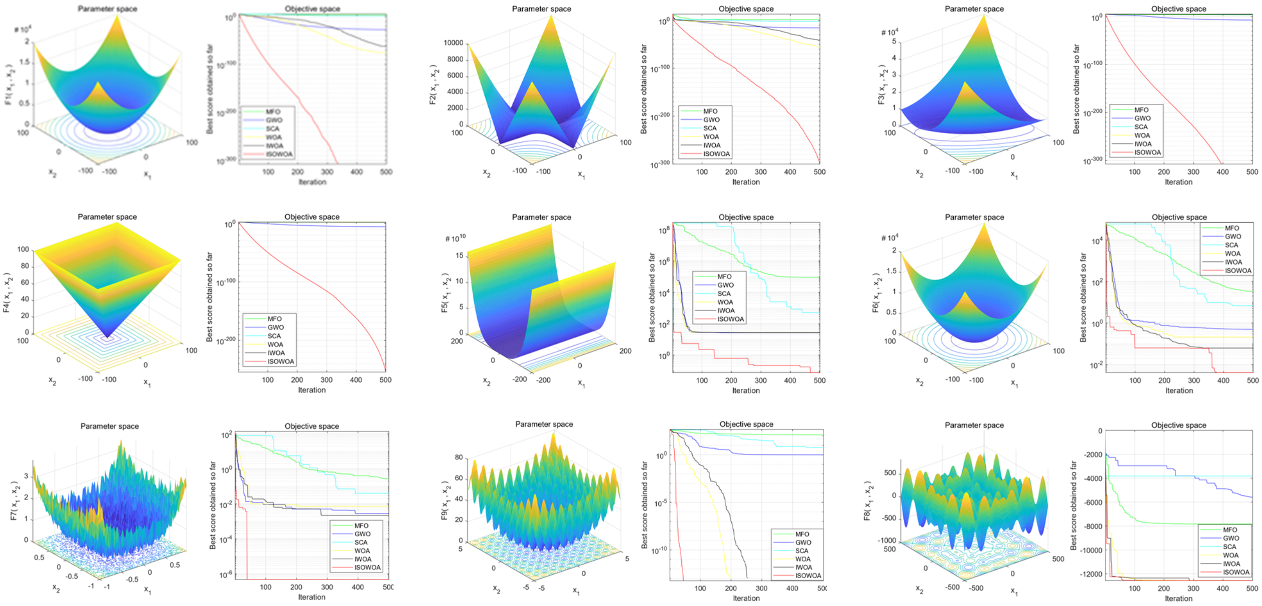

Figure 4. Convergence behavior of tested algorithm (MFO = moth flame optimization, GWO = grey wolf optimizer, SCA = sine cosine algorithm, IWOA = improved whale optimization algorithm， ISOWOA = improved selective opposition whale optimization algorithm.

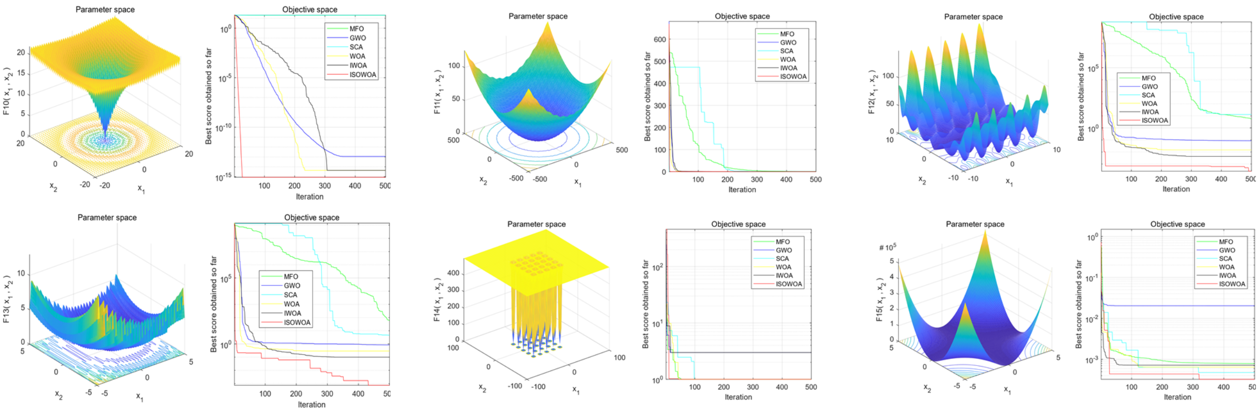

Figure 5. Convergence behavior of tested algorithm.

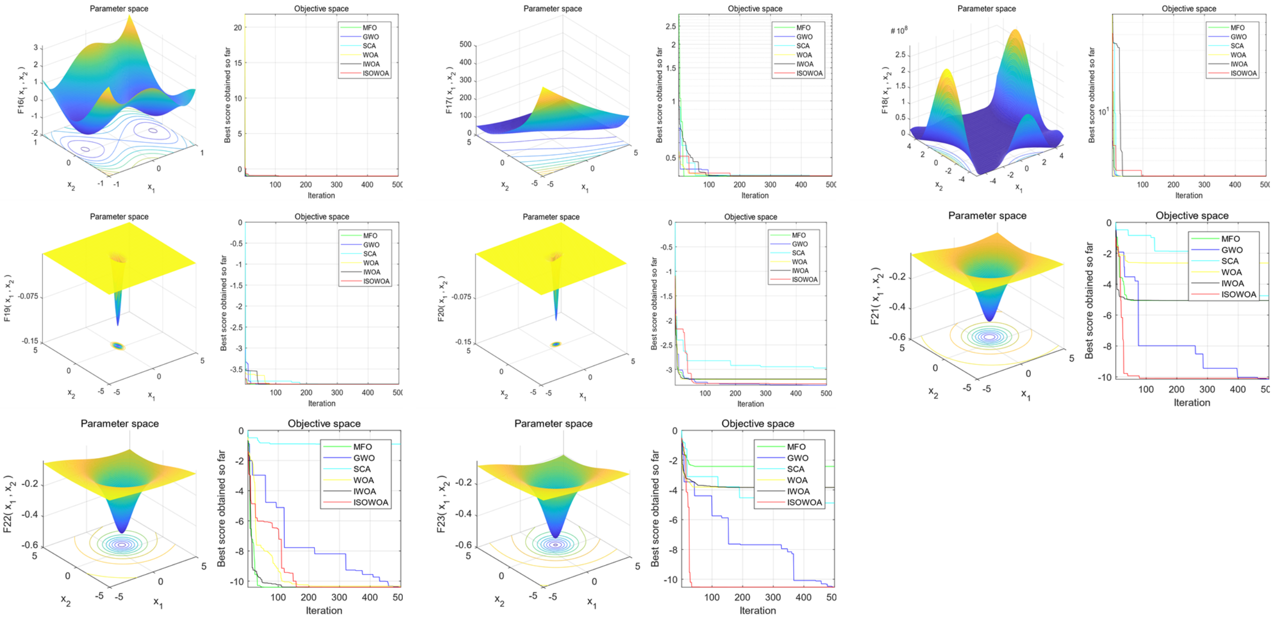

Figure 6. Convergence behavior of tested algorithm.

It is not difficult to see from Figure 6 that the fixed-dimension F18–F23 functions can obtain the minimum value of functions with the smallest number of iterations compared with other variant algorithms, which once again proves the powerful optimization efficiency of ISOWOA. It is noted that F20 and F22 do not show optimal performance among all variant algorithms in early iterations, but it is clear from the image that the enhanced algorithm has powerful searching ability to achieve the best optimization performance in the middle and late stages. Therefore, it can be seen that ISOWOA is a balanced algorithm with better global and local search capabilities.

4.6. Function test results of multiple dimensions

The optimized ISOWOA was applied to the tests of the three dimensions of 100, 500, and 1000 to ensure that the multi-dimensional function could run independently and had a certain stability.

Table 6. Test results of 100 dimensional unimodal functions (F1–F7).

Function	Criteria	ISOWOA	JSWOA	AWOA	OWOA	OBCWOA	LWOA	WOA
F1	ave	**0**	0	0	1.49E−173	0	4.25E-91	1.21E-73
	std	**0**	0	0	0	0	3.00E-90	3.63E-73
F2	ave	**0**	0	1.07E−178	1.46E−137	0	4.41E-73	2.91E-50
	std	**0**	0	0	7.78E−137	0	3.05E-72	8.97E-50
F3	ave	**0**	0	0	1.27E+06	0	1.86E+05	1022126.4
	std	**0**	0	0	4.59E+05	0	2.38E+04	214442.09
F4	ave	**0**	**0**	1.89E−169	8.90E+01	**0**	1.84E+01	77.4423
	std	**0**	0	0	1.22E+01	0	1.19E+01	20.7682
F5	ave	**4.68E-02**	9.81E+01	9.81E+01	9.80E+01	9.71E+01	9.80E+01	98.2389
	std	1.17E-01	1.40E−01	8.90E−02	2.83E−01	6.02E-01	**2.66E-02**	0.1572
F6	ave	**1.04E-02**	3.95E+00	1.99E+00	5.12E+00	4.41E+00	8.37E-01	3.859
	std	**1.48E-02**	9.34E−01	8.31E−01	1.12E+00	1.33E+00	1.42E-01	1.1369
F7	ave	8.87E-05	6.90E-05	8.93E−05	5.24E−04	**6.10E-05**	4.46E-03	0.0037956
	std	6.31E-05	5.53E−05	8.92E−05	1.06E−03	**5.98E-05**	5.12E-03	0.0053394

The test results show that ISOWOA can solve optimization problems of large dimensions without serious dimensional disasters. Since no corresponding data are listed in the original literature [19], the WOA data in Table 6 are current environmental test results. EWOA does not show super-scale optimization results; so, we selected results in the corresponding literature of JSWOA [35] and OBCWOA [39]. Tables 6–11 shows the horizontal comparison results.

Table 7. Test results of 100 dimensional multimodal functions (F8–F13).

Function	Criteria	ISOWOA	JSWOA	AWOA	OWOA	OBCWOA	LWOA	WOA
F8	ave	**-41890.91**	−4.16E+04	−4.14E+04	−4.14E+04	-3.32E+04	-3.83E+04	-34501.1
	std	**11.1604**	1.05E+03	1.19E+03	7.85E+02	6.02E+03	4.47E+03	5674.5881
F9	ave	**0**	0	0	0	0	4.552E-15	**0**
	std	**0**	0	0	0	0	2.251E-14	**0**
F10	ave	**8.88E-16**	**8.88E-16**	**8.88E-16**	2.43E−15	**8.88E-16**	3.73E-15	4.56E-151
	std	**0**	0	0	2.02E−15	0	2.38E-15	2.55E-152
F11	ave	**0**	0	0	0	0	6.53E-03	0.0122543
	std	**0**	0	0	0	0	4.62E-02	0.0671164
F12	ave	**5.95E-05**	7.13E−02	2.45E−02	8.19E−02	6.43E-02	9.31E-03	0.0514455
	std	**6.23E-05**	2.51E−02	1.45E−02	2.69E−02	3.33E-02	2.53E-03	0.0223976
F13	ave	**1.59E-03**	2.36E+00	1.07E+00	4.04E+00	3.62E+00	7.891E-01	3.1365
	std	**2.48E-03**	5.79E−01	5.44E−01	1.40E+00	1.13E+00	2.631E-01	1.086

From Tables 6 and 7, we can see that, in the 100-dimension function test, the horizontal comparison with the results in the references shows that ISOWOA still maintains its original stability without the occurrence of dimensional disasters, and results of the functions remain near the theoretical extreme value.

In view of F5–F7, the mean value of ISOWOA is also one or three orders of magnitude better than that of modern metaheuristic algorithm and variants of WOA.

Table 8. Test results of 500 dimensional unimodal functions (F1–F7).

Function	Criteria	ISOWOA	IWOA	AWOA	OWOA	OBCWOA	LWOA	WOA
F1	ave	**0**	0	0	2.2E−171	0	5.59E-95	3.76E-69
	std	**0**	0	0	0	0	2.46E-94	1.46E-68
F2	ave	**0**	0	3.5E−179	2.7E−137	0	2.82E-73	1.29E-48
	std	**0**	0	0	1.48E−13	0	1.84E-72	4.64E-48
F3	ave	**0**	0	0	3.01E+07	0	5.79E+6	3.42 E+7
	std	**0**	0	0	1.16E+07	0	8.25E+5	1.03 E+7
F4	ave	**0**	**0**	4.1E−170	7.54E+01	**0**	5.06E+1	54.1912
	std	**0**	0	0	3.37E+01	0	2.34E+1	27.2712
F5	ave	**0.19283**	4.95E+02	4.95E+02	4.96E+02	4.92E+2	4.94E+2	496.2001
	std	0.31447	**2.03E−01**	3.25E−01	5.35E−01	4.00E-01	6.08E-01	0.37054
F6	ave	**3.90E-02**	2.47E+01	1.15E+01	4.03E+01	4.36E+1	8.31E+0	28.2923
	std	**4.77E-02**	5.64E+00	4.93E+00	8.33E+00	9.22E+0	1.00E+0	8.8914
F7	ave	1.10E-04	1.02E−04	7.58E−05	3.58E−04	**6.17E-05**	4.01E-3	0.003300
	std	9.70E-05	8.16E−05	6.07E−05	5.46E−04	**5.74E-05**	3.96E-3	0.003975

Table 9. Test results of 500 dimensional multimodal functions (F8–F13).

Function	Criteria	ISOWOA	IWOA	AWOA	OWOA	OBCWOA	LWOA	WOA
F8	ave	**-209434**	−2.09E+5	−2.08E+5	−2.08E+5	-1.61E+05	1.90E+05	170759.7
	std	**80.4923**	1.95E+03	3.19E+03	3.39E+03	2.92E+04	2.51E+04	29638.545
F9	ave	**0**	0	0	0	0	7.28E-14	3.03E-14
	std	**0**	0	0	0	0	2.49E-13	1.66E-13
F10	ave	**8.88E-16**	**8.88E-16**	**8.88E-16**	2.19E−15	**8.88E-16**	3.80E-15	4.32E-15
	std	**0**	0	0	2.38E−15	0	2.45E-15	2.38E-15
F11	ave	**0**	0	0	3.70E−18	0	2.22E-18	**0**
	std	**0**	0	0	3.70E−18	0	1.57E-17	**0**
F12	ave	**2.28E-05**	7.14E−02	2.17E−02	1.43E−01	5.87E-01	1.33E-02	0.076393
	std	**3.27E-05**	2.21E−02	1.30E−02	5.60E−02	1.18E-01	2.13E-03	0.035297
F13	ave	**5.38E-03**	1.37E+01	5.16E+00	1.94E+01	4.78E+01	4.42E+00	19.6528
	std	**9.48E-03**	2.51E+00	1.87E+00	3.52E+00	5.41E+00	6.75E-01	7.0613

From the results of 500-dimension functions in Tables 8 and 9, we can see that the valley function still shows a good effect. Although the order of magnitude has decreased precision, compared with other algorithms, it is still closest to the theoretical value.

Table 10. Test results of 1000 dimensional unimodal functions (F1–F7).

Function	Criteria	ISOWOA	JSWOA	AWOA	OWOA	WOA
F1	ave	**0**	0	0	2.98E−172	1.81E-68
	std	**0**	0	0	0	9.89E-68
F2	ave	**0**	0	5.41E−178	1.53E−140	1.93E-48
	std	**0**	0	0	8.25E−140	5.69E-48
F3	ave	**0**	0	0	1.25E+08	132211500
	std	**0**	0	0	3.93E+07	48442106.1
F4	ave	**0**	**0**	1.85E−174	7.98E+01	63.782
	std	**0**	0	0	2.61E+01	23.0503
F5	ave	**1.09E+00**	9.91E+02	9.90E+02	9.92E+02	993.8017
	std	2.51E+00	**4.08E−01**	6.26E−01	7.01E−01	0.70666
F6	ave	**1.39E-01**	4.46E+01	2.12E+01	7.50E+01	72.8772
	std	**1.99E-01**	9.32E+00	8.76E+00	1.59E+01	19.5057
F7	ave	1.60E-04	8.78E−05	**8.27E−05**	7.74E−04	0.0035431
	std	8.44E-05	7.00E−05	**6.63E−05**	2.06E−03	0.0044305

Table 11. Test results of 1000 dimensional multimodal functions (F8–F13).

Function	Criteria	ISOWOA	JSWOA	AWOA	OWOA	WOA
F8	ave	**-418948**	−4.17E+05	−4.12E+05	−4.16E+05	-335844.27
	std	**72.2583**	5.03E+03	1.55E+04	6.66E+03	62629.144
F9	ave	**0**	0	0	0	1.82E-13
	std	**0**	0	0	0	5.55E-13
F10	ave	**8.88E−16**	**8.88E−16**	**8.88E−16**	3.85E−15	4.91E-15
	std	**0**	0	0	2.81E−15	2.59E-15
F11	ave	**0**	0	0	0	3.70E-18
	std	**0**	0	0	0	2.03E-17
F12	ave	**2.82E-05**	7.15E−02	2.50E−02	1.34E−01	0.10518
	std	**4.10E-05**	2.59E−02	1.55E−02	4.32E−02	0.062805
F13	ave	**1.13E-02**	2.56E+01	1.05E+01	4.35E+01	36.1263
	std	**1.70E-02**	5.85E+00	5.37E+00	1.09E+01	13.0893

We can see that, F1–F3 functions still maintain the theoretical extreme value in the 1000-dimension test. It is worth mentioning that F5 and F6 functions still maintain their precision on the basis of surpassing other metaheuristic algorithms. It can be seen from Tables 10 and 11 that the powerful optimization ability of the ISOWOA can greatly make F1–F4, F9 and F11 find the optimal value. The convergence accuracy of ISOWOA is still better than that of the other four algorithms in multi-dimensional function tests. On the whole, results of 39 multi-dimensional function tests exceed other algorithms except F7. These tests showed that ISOWOA is superior at solving large-scale multi-dimensional optimization problems and maintaining a certain stability while avoiding serious dimensional disasters. In summary, ISOWOA has remarkable performance in addressing complex higher dimensional functions. Because of this, this paper considers it can be greatly applied to deal multi-constraints in DNA storage.

4.7. Wilcoxon's rank sum test and Friedman's nonparametric rank test

Table 12. Wilcoxon's rank sum test of algorithms.

Function	ISOWOA	EWOA	JSWOA	WOA	AGWO	AWOA
F1	1	3	3	5	6	3
F2	1	3	2	5	6	4
F3	1	3	3	5	6	3
F4	1.5	3	1.5	6	5	4
F5	1	2	4	5	3	6
F6	1	2	5	3	6	4
F7	1	4	2	6	5	3
F8	1	2	3	5	6	4
F9	2	2	4.5	2	6	4.5
F10	2.5	2.5	2.5	5	6	2.5
F11	1.5	1.5	3.5	6	5	3.5
F12	1	2	5	4	6	3
F13	1	2	5	4	6	3
F14	1	2	6	5	4	3
F15	1.5	5	3	4	6	1.5
F16	6	2	4.5	2	2	4.5
F17	6	1.5	3	1.5	4	5
F18	5	6	1.5	3	1.5	4
F19	3.5	6	3.5	2	1	5
F20	1	6	4	2	3	5
F21	1	3	2	4	5	6
F22	1.5	3	1.5	5	4	6
F23	1	3	2	5	4	6
Sum	**44**	69.5	75	94.5	106.5	93.5
rank	**(1)**	(2)	(3)	(5)	(6)	(4)

Wilcoxon and Friedman tests were employed for horizontal and vertical comparisons of the mean value of ISOWOA. We used statistical method (Wilcoxon's rank-sum and Friedman's nonparametric rank) to validate the superiority and significance of the ISOWOA and to investigate the difference between our algorithm and other improved algorithms, respectively. Because there are three sets of missing data in OBCWOA [39] and LWOA [39], we only selected the other five algorithms for significance analysis. The order of each algorithm using the rank-sum test is shown in Table 12.

It can be seen from Table 12 that the rank means of the six algorithms are 1.91 (ISOWOA), 3.02

(EWOA), 3.26 (JSWOA), 4.11 (WOA), 4.63 (AGWO), and 4.07 (AWOA). The priority order is ISOWOA > EWOA > JSWOA > AWOA > WOA > AGWO. We see that ISOWOA ranks first, with an obvious horizontal comparative advantage, which demonstrates its superiority and significance. Table 13 lists the test results of all benchmark functions after vertical comparison, based on analysis using IBM SPSS Statistics 26 software. Based on a large number of statistical data, the significance level was set as 0.05. The confidence interval level (CILEVEL) was set to 95%. SPSS calculated the statistics and corresponding P-values. If P is less than 0.05, two algorithms can be considered to be significantly different, and there is no statistically significant difference when P is greater than 0.05.

Hence, a small P indicates a better solution effect, i.e., the tests of our optimization algorithm have a qualitative breakthrough compared with those of modern metaheuristic algorithm and variants of WOA. Table 13 lists the P-values. Where N/S/E/I represent the overall statistical quantitative analysis that ISOWOA is superior (S), equal (E) and inferior (I) compared with other algorithms in N problems.

Table 13. P-values using Friedman's nonparametric test for functions F1–F23.

Comparison	p-value	N/S/E/I
ISOWOA vs. EWOA	0.012	23/18/3/2
ISOWOA vs. WOA	0.002	23/20/1/2
ISOWOA vs. AWOA	0.02	23/21/2/0
ISOWOA vs. JSWOA	0.052	23/17/3/3
ISOWOA vs. AGWO	0.018	23/20/0/3

We see that ISOWOA had more signification compared to other algorithms. Among them, P = 0.012 when comparing ISOWOA and EWOA, indicating a statistically significant difference. In addition, compared with AWOA, the algorithm proposed in this paper has a dominant ratio of 91.3%, which is not 100% because the function has reached the theoretical extreme value and cannot be further optimized. These results conform to the intuitive results in Tables 3–5.

4.8. Performance of ISOWOA algorithm on CEC-2015 test function set

Table 14. Average fitness value obtained by algorithms on IEEE CEC-2015 function.

Dim	Algorithms	CEC-1	CEC-2	CEC-3	CEC-4	CEC-5
30	IWOA	3.80E+07	5.58E+08	**2.01E+01**	2.49E+02	4.28E+03
	MFA	9.85E+04	3.74E+03	2.09E+01	5.15E+01	2.87E+03
	ISOWOA	**8.97E+04**	**2.70E+03**	**2.01E+01**	**5.13E+01**	**2.39+03**
50	IWOA	1.06E+08	4.20E+09	2.03E+01	5.51E+02	7.15E+03
	MFA	8.38E+05	1.35E+04	2.11E+01	**1.35E+02**	5.02E+03
	ISOWOA	**8.35E+05**	**7.96E+03**	**2.01E+01**	**1.35E+02**	**4.90E+03**
90	IWOA	3.45E+08	1.62E+11	2.08E+01	1.54E+03	2.00E+04
	MFA	6.50E+06	**7.13E+03**	2.13E+01	3.73E+02	1.27E+04
	ISOWOA	**6.21E+06**	8.01E+03	**2.05E+01**	**3.29E+02**	**1.13E+04**

Table 15. Average fitness value obtained by algorithms on IEEE CEC-2015 function.

Dim	Algorithms	CEC-6	CEC-7	CEC-8	CEC-9	CEC-10
	IWOA	1.37E+06	1.91E+01	6.46E+05	1.90E+02	2.97E+06
30	MFA	**3.98E+04**	**4.65E+00**	**2.40E+04**	**1.02E+02**	1.95E+04
	ISOWOA	1.06E+05	4.98E+00	3.79E+04	**1.02E+02**	**1.78E+04**
	IWOA	2.84E+06	1.20E+02	2.84E+06	3.63E+02	4.50E+06
50	MFA	**2.46E+05**	**3.81E+01**	**1.52E+05**	**1.04E+02**	**6.73E+03**
	ISOWOA	4.96E+05	4.38E+01	2.14E+06	**1.04E+02**	2.03E+04
	IWOA	2.12E+07	4.66E+02	1.05E+07	1.05E+03	2.30E+07
90	MFA	**1.54E+06**	**1.28E+02**	**6.73E+05**	**1.08E+02**	**7.63E+03**
	ISOWOA	4.47E+06	2.79E+02	2.46E+06	1.37E+02	3.49E+05

Table 16. Average fitness value obtained by algorithms on IEEE CEC-2015 function.

Dim	Algorithms	CEC-11	CEC-12	CEC-13	CEC-14	CEC-15
	IWOA	8.64E+02	1.46E+02	6.12E-02	3.64E+04	1.15E+02
30	MFA	**4.32E+02**	1.04E+02	2.64E-02	3.30E+04	**1.00E+02**
	ISOWOA	4.46E+02	**1.02E+02**	**1.98E-02**	**2.08E+04**	**1.00E+02**
	IWOA	1.76E+03	1.81E+02	3.45E-01	8.53E+04	1.15E+02
50	MFA	**8.13E+02**	1.54E+02	8.58E-02	6.51E+04	1.00E+02
	ISOWOA	1.03E+03	**1.47E+02**	**7.42E-02**	**5.96E+04**	**1.00E+02**
	IWOA	4.00E+03	2.03E+02	5.39E-01	2.21E+05	1.51E+04
90	MFA	**2.03E+03**	1.55E+02	6.48E-02	**1.09E+05**	**1.03E+02**
	ISOWOA	3.21E+03	**1.52E+02**	**4.27E-02**	**1.09E+05**	1.08E+02

In order to better demonstrate the proposed ISOWOA, a set of 15 real-parameter single-objective optimization problems of IEEE CEC 2015 (CEC-1-CEC-15) is considered. The efficiency of the ISOWOA is compared against improved whale optimization algorithm (IWOA) [40] and modified firefly algorithm [41]. For minimum interference, each algorithm is executed for 51 runs over three dimensions, i.e., 30, 50, and 90. The best results for all tests are in bold in Tables 14–16.

When executed on 30 and 50 dimensions, it can be clearly seen from Tables 14–16 that the proposed ISOWOA is superior to the comparison algorithm in CEC-1, CEC-2, CEC-3, CEC-4, CEC-5, CEC-9, CEC-10, CEC-12, CEC-13, CEC-14 and CEC-15 benchmark functions. In addition, from any dimension, the results obtained by the proposed algorithm on CEC-1, CEC-3, CEC-4, CEC-5, CEC-12, CEC-13 and CEC-14 functions are always equal to or exceeded all other algorithms. To sum up, the proposed ISOWOA also performs the compared algorithms on most benchmark problems. Therefore, the test function set of CEC-2015 proves the superiority of ISOWOA algorithm again.

According to the analysis of the above experimental data, through the verification of CEC 2005 and CEC 2017 standard test function sets of different dimensions and scales, it can be concluded that the ISOWOA algorithm proposed in this paper has greatly improved the convergence effect compared with the original algorithm and other similar intelligent optimization algorithms. Among them, the average value obtained by ISOWOA algorithm in the unimodal function is closer to the theoretical optimal value, and the standard deviation is stable, which indicates that the improved algorithm has stronger exploration ability and makes the result closer to the global optimal value. In the multimodal function, both the function value and the standard deviation reach the minimum, which indicates that

ISOWOA algorithm still has excellent optimization ability in the limited search space, and reflects that the algorithm proposed in this paper has better stability while ensuring higher performance. In addition, in the composite function, the improved algorithm also has different degrees of competitive advantages, which shows that the algorithm also has robustness in solving global optimization problems. The above indicates that the proposed algorithm is a satisfactory optimization algorithm with good performance and stability.

4.9. Computational complexity analysis of the ISOWOA algorithm

The computational complexity of ISOWOA algorithm includes the following three main parts:

1) Population initialization. The computational complexity of this part depends on the population size N and the population dimension D, which generally does not exceed O (N*D). The ISOWOA proposed in this paper can be compared with other algorithms, which is O (N*D).

2) Calculation of initial population fitness. The computational complexity of this part depends on the population size N and the target fitness F, which generally does not exceed O (N*F). The ISOWOA proposed in this paper can be compared with other algorithms, which is O (N*F).

3) In the main cycle, the computational complexity of this part depends on the number of iterations T, the population size N, the population dimension D and the target fitness F, which generally does not exceed O (T*N*D + T*N*F).

The computational complexity of the algorithm proposed in this paper depends on the population location update and fitness calculation. The population location update is determined to adopt any of the three predator-prey methods through the cycle judgment conditions. As shown in the figure, when the algorithm meets the execution conditions, the left branch is executed, that is, randomly searching prey and surrounding prey. When the algorithm does not meet the execution conditions, it executes the right branch, that is, bubble prey. The computational complexity of this part does not exceed O (T*N*D), and the calculation of fitness does not exceed O (T * N * F). Therefore, the computational complexity of the main cycle can be expressed as O (T*N*D + T * N * F), which can also be compared with other algorithms. From the above analysis, we can conclude that the time complexity of the proposed ISOWOA algorithm is equivalent to that of other algorithms.

In general, the computational complexity of the ISOWOA algorithm proposed in this paper is as follows: O (ISOWOA) = O (Initialization whale population) + T * O (Estimate the fitness of whales) + T*O (Update the position of all whales). Where O (Steps to initialize whale population) = O (N), T*O (steps to estimate the fitness of whales) = T*O (N)， and T*O (Steps to update the position of all whales) = T*O (N*D). So, the total time complexity is O (ISOWOA) = O (N) + T*O (N) + T*O (N*D) = O (T*N*D).

5. Applications in building DNA storage codes

5.1. DNA coding

DNA coding refers to mapping a binary string into a corresponding base sequence by a certain coding method, that is, a sequence composed of A, T, C and G. Recently a group of new concepts were designed for DNA codes. DNA coding is used in technologies such as DNA computing [42], DNA nanostructures, data encryption, data storage, and DNA microarray technology and image encryption [43]. In these applications, information is always represented by a unique sequence of DNA, and processing is accomplished by hybridization between these DNA codes. The quality of the codes directly determines the success or failure of the application. Recently a group of new concepts were designed for DNA codes. Hong et al. [44] used theory to construct DNA coding sets and attempted to construct them by irreducible cyclic codes. Wang et al. [45] designed a new image encryption scheme combined with DNA encoding using chaotic sequence and diffusion operations of a DNA matrix, which can resist common attack methods. Song et al. [46] proposed a conversion method to achieve a storage rate of 1.9 bit/base. This method greatly meets the requirements of low encoding complexity.

For the DNA coding problem, the current research content is mainly divided into two aspects: coding quality and coding quantity. When solving practical problems, we need to choose the coding direction according to the focus of the problem: 1) Under the condition of sufficient coding quantity, the research goal is to improve the coding quality; 2) Under the premise that the coding quality has reached the requirements, the research goal is to improve the number of codes. In general, quality optimization and the number of DNA codes often fail to achieve a win-win situation. There is no good way to solve this NP-complete problem because the number of DNA codes always decreases with increasing constraints. Recently, Nguyen et al. [47] designed constrained codes with error-correction capability, which were capable of correcting a single insertion / edited. wang et al. [48] presented novel similarity significance (SS) model to measure the similarity between DNA sequences. With this model, they constructed thermodynamically stable DNA codes using a sorting-based algorithm. The coding quality is improved at the expense of the number of codes. It has strong limitations in practical applications, and only meets certain specific scenarios without considering the DNA storage coding rate. In addition, Cao et al. [49] improved the DNA storage coding rate by improving the MVO algorithm. Under the fixed sequence length, more DNA sequences were screened out, which reduced the storage cost to a certain extent. Limited by the performance of the algorithm, it cannot achieve a big breakthrough in the lower bound of the DNA storage coding set. How-ever, WOA algorithm has the advantage of strong global search ability in discrete problems. Under the premise of ensuring sequence quality, it can effectively improve the number of DNA coding storage coding sets. Based on this, WOA was introduced to DNA computing to construct DNA storage codes. The ISOWOA algorithm with three novel strategies can search for codes that meet the constraints in a wider range, realizing the construction of a larger number of coding sets within a certain base length, The in-crease in the size of the coding set means that shorter DNA strands can be used to store more data, which greatly improves the coding rate of storage and reduces the cost of data storage.

For the construction of DNA storage codes, we aim to further improve the size of coding set and the coding rate under the same constraint requirements [49]. According to the characteristics of DNA storage coding set, this paper models the combination constraint as a multi-objective problem of the algorithm, and converts the DNA sequence from A, T, C, G to 0, 1, 2, 3. Secondly, in order to match the transformation mechanism, the original continuous whale optimization algorithm is discretized. Finally, the improved ISOWOA is applied to screen the DNA storage coding sets that meet the four constraints of uncorrelated address, edit distance, no run length, and GC content.

5.2. Constraints of DNA storage coding set

The constraints of DNA coding sets are categorized as combinatorial [50], application-oriented, and thermodynamic [51]. Combinatorial constraints include Hamming distance (HD), GC-content, forbidden constraints, and so on. Application-oriented constraints include uncorrelated and run-length constraints. Thermodynamic constraints include free energy, melting temperature, and minimum energy constraints. Our experiments have shown that reliable DNA codes designed with uncorrelated address, edit distance, no run-length, and GC-content. The experimental results show that the number of multi-constraint storage codes with different lengths constructed in this paper increases by 2–18%, reaching or exceeding the previous optimal construction [49].

5.2.1. Editing distance constraint

The minimum edit distance (MED), or levenshtein distance, is the minimum operand required to convert one string to another. We set the minimum editing distance [49] as a similarity calculation function, which is used for the construction of DNA storage codes. There are two strings in V, and $u\ and\ v$ satisfy

$$\eta(u, v) \geq d. \tag{10}$$

where d is an integer, and η is the set of editing distance constraints for DNA sequences.

By introducing editing distance to the construction of a DNA storage code, problems, such as substitution, insert, and delete in nanopore sequencing technology, are well solved. Sequencing errors are minimally avoided. $G(u, v)$ represents the editing distance between u and v. The objective function is

$$UE(u_i) = \min_{1 \leq j \leq n, j \neq i} \{G(u_i, v_j)\} \geq d. \tag{11}$$

where $\mathrm{UE}(u_i)$ defines the minimum $G(u_i, v_j)$.

5.2.2. GC-content constraint

DNA computing requires stable and uniform DNA molecules, composed of A, T, C, and G bases. GC content is the ratio of guanine and cytosine in four DNA bases in a DNA sequence, and the content constraint must meet the range in chemical determination, generally $40 \leq GC \leq 60$. Thus, we set a standard of 50 for the sequence of GC-content.

For a sequence of length s, $GC(s)$ can be expressed as [49]:

$$GC(s) = \frac{|G+C|}{|s|}. \tag{12}$$

where $|G + C|$ is the fixed number that are either G or C within each code.

5.2.3. Sub-subheading

Run-length (RL) is the length formed by the repeated occurrence of each base in a data stream composed of sequences. Given the nature of DNA molecular synthesis and sequencing, repeated bases may lead to the failure of an experiment. For example, in ACGGGT, G is repeated; this sequencing can be misinterpreted as ACGT or ACGGT, which will lead to reading and writing errors. The constraint is defined as [49]

$$S_i \neq S_{i+1} \qquad i \in [1, n-1]. \tag{13}$$

where S_i is a DNA sequence $S(S_1, S_2, S_3, \ldots, S_n)$ of length n.

5.2.4. Uncorrelated address constraint

In computer science, basic units of memory used for data access are assigned unique serial numbers, also known as memory addresses, which refer to specific locations in a system. In DNA computing, to recover data information stored by DNA, molecules in the reaction pool must be prefixed with a specific address. Address sequences must not be similar, to avoid failed retrieval of data block information. Uncorrelated address constraints are defined as follows.

In sequences $L(l^1, l^2, l^3, \ldots l^s)$ and $H(h^1, h^2, h^3, \ldots, h^n)$, the suffix of L cannot appear as the prefix of H, and vice versa [52].

Assuming that we define an uncorrelated address constraint distance as 3, then CTAGAT and CGACTA cannot occur together in the entire sequence pool, because CTA is a related character [49]. The coding set B obtained by the uncorrelated address constraint can not only eliminate cross-hybridization between addresses but can also avoid sequence selection errors in the sequencing process.

5.3. Bounds on DNA storage codes

The steps of constructing DNA storage codes are as follows:

1) The required parameters and DNA sequence are initialized. Then, WOA is discretized. By evaluating their fitness, current optimal solution is screened out, and DNA sequences satisfying the constraint are incorporated to form the first-generation population.

2) The populations are sorted in order of fitness, by using the correlation to select the individuals and dimensions to be opposed.

3) The updated population is involved in iterative optimization, and the optimized result is taken as the input to determine whether the constraint conditions are met. If so, it is incorporated, otherwise iterative optimization is continued.

4) Finally, when the maximum number of iterations is reached, the DNA storage coding set is outputted.

Superscripts are presented in Table 17, where d represents DMVO [49] and i represent ISOWOA. In Table 18, $A^{GC,NL,UA}(n,d)$ represents a DNA coding set, which satisfies the editing distance, GC-content, no run-length, and uncorrelated address constraints. This value for $A^{GC,NL,UA}(n,d)$ represents the maximum number of DNA storage codes constructed by ISOWOA, satisfying the same constraints, where d represents the storage editing distance, and n represents the length of the DNA storage coding. From the perspective of biological computing, we aim to screen out more DNA storage codes with the same combinatorial constraints.

In fact, the combinatorial constraints applied in this paper can be regarded as a multi-objective application. Through the previous tests, it is verified that the ISOWOA proposed in this paper has strong equilibrium ability in solving multi-objective optimization problems. Because of this, ISOWOA is introduced as a method to construct the DNA storage coding set, and the reliability of the algorithm in practical application is further discussed.

Table 17. Meanings of superscripts.

Superscript	Meaning
u	DMVO algorithm
i	ISOWOA algorithm

Table 18. Lower bounds of different lengths satisfying constraints ($\boldsymbol{A^{GC,NL,UA}(n,d)}$).

n\d	3	4	5	6	7	8	9	10
4	6^{u} $\mathbf{6^{i}}$							
5	12^{u} $\mathbf{14^{i}}$	5^{u} $\mathbf{5^{i}}$						
6	30^{u} $\mathbf{31^{i}}$	11^{u} $\mathbf{12^{i}}$	4^{u} $\mathbf{4^{i}}$					
7	53^{u} $\mathbf{56^{i}}$	19^{u} $\mathbf{19^{i}}$	6^{u} $\mathbf{7^{i}}$	3^{u} $\mathbf{3^{i}}$				
8	101^{u} $\mathbf{108^{i}}$	38^{u} $\mathbf{43^{i}}$	12^{u} $\mathbf{12^{i}}$	5^{u} $\mathbf{5^{i}}$	3^{u} $\mathbf{3^{i}}$			
9	167^{u} $\mathbf{180^{i}}$	58^{u} $\mathbf{67^{i}}$	19^{u} $\mathbf{20^{i}}$	7^{u} $\mathbf{7^{i}}$	3^{u} $\mathbf{3^{i}}$	2^{u} $\mathbf{2^{i}}$		
10	250^{u} $\mathbf{267^{i}}$	110^{u} $\mathbf{118^{i}}$	34^{u} $\mathbf{38^{i}}$	11^{u} $\mathbf{13^{i}}$	5^{u} $\mathbf{5^{i}}$	3^{u} $\mathbf{3^{i}}$	2^{u} $\mathbf{2^{i}}$	

In Table 18, we see that under the same constraints, the number of different length coding sets constructed by ISOWOA all equaled or exceeded DMVO algorithm. The lower bounds generally increased by 2–18%, which shows that our work has been significantly improved compared with that of our predecessors. If the sequence length is equal to 5, 6, 7, 8, 9, 10, the coding lower bound of the proposed algorithm has been broken through. It is worth noting that if $n = 10$ and $d = 6$, the number of DNA storage codes in our result is 18% higher than that in the optimal result of DMVO, meeting the evaluation index of the DNA sequence coding set.

In addition, the experimental data also show that the advantages of the algorithm are more significant with the increase of sequence length. When the editing distance constraint value is defined as 3, we can find that the lower bounds of DNA sequence length of 7, 8 and 9 constructed by the algorithm are increased by 5.7, 7 and 8% respectively. And the corresponding coding rate $R = log^{\wedge}4M/n$ is also increased by more than 1%. If d = 3, our coding rate at n = 9 (0.416) exceeds that at n = 10 (0.403), which also means that the algorithm can effectively reduce the synthesis cost of sequences under the same performance requirements. As expected, the enhanced algorithm shows good performance. The results confirm the applicability of ISOWOA to biological computing.

Finally, the non-payload DNA storage coding set after selection can avoid insertion, deletion, and replacement errors to the greatest extent in the process of nanopore sequencing. Based on this storage coding set, access to large DNA files can be successfully completed in the fourth generation of nanopore sequencing.

6. Conclusions

In dealing with optimization problems, the whale optimization algorithm has the drawback that local optimal solutions are prone to precocious convergence, i.e., it is assumed to be the best solution when the global optimal solution has not been found. Based on this, the proposed algorithm integrates the strategies of enhanced quasi-opposition learning, a nonlinear convergence factor, and a new time-varying inertia weight with collaborative convergence factor, giving it a stronger search ability in the whole algorithm optimization stage. The adaptability and convergence precision of each iteration are improved, and the inherent defect of premature convergence is eliminated, so that it becomes an equilibrium algorithm. Through 38 benchmark functions, the proposed algorithm was compared to modern metaheuristic algorithm and variants of WOA on test cases based on multiple dimensions, number of iterations, mean value, and variance. The experiment shows that the results of 17 functions of CEC 2005 obtained by ISOWOA were obviously better than those of EWOA in terms of mean and standard deviation, and the results of another four functions were the same. In addition, the algorithm is superior to other algorithms in 11 functions in the test function set of CEC 2015, which again proves the superiority of ISOWOA algorithm. In biological computing, based on the same combinational constraints, the multi-constraint DNA storage coding set constructed by ISOWOA has a better solving effect than DMVO. The results show that the number of DNA storage codes filtered by ISOWOA increased by 2–18%, and the coding set can realize access to DNA files in biological experiments.

In future work, we will take DNA storage coding as the main research direction and apply various nature-inspired algorithms to address optimization problems, such as machine learning, engineering problems, networks, and benchmark functions.

Acknowledgments

This work is supported by the National Key Technology R&D Program of China (No. 2018YFC0910500), the National Natural Science Foundation of China (Nos. 61425002, 61751203, 61772100, 61972266, 61802040), LiaoNing Revitalization Talents Program (No. XLYC2008017), the Innovation and Entrepreneurship Team of Dalian University (No.XQN202008), Natural Science Foundation of Liaoning Province (Nos. 2021-MS-344, 2021-KF-11-03, 2022-KF-12-14).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, Q. Zhang, A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times, *IEEE Trans. Cybern.*, **51** (2021), 1430–1442. https://doi.org/10.1109/TCYB.2019.2939219
2. W. Deng, J. Xu, Y. Song, H. Zhao, Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem, *Appl. Soft Comput.*, **100** (2021), 1568–4946. https://doi.org/10.1016/j.asoc.2020.106724

3. F. Zhao, X. He, L. Wang, A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem, *IEEE Trans. Cybern.*, **51** (2020), 5291–5303. https://doi.org/10.1109/TCYB.2020.3025662
4. A. Kaveh, A. D. Eslamlou, Water strider algorithm: A new metaheuristic and applications, *Structures*, **25** (2020), 520–541. https://doi.org/10.1016/j.istruc.2020.03.033
5. F. Zhao, L. Zhang, J. Cao, A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem, *Comput. Ind. Eng.*, **153** (2021), 107082. https://doi.org/10.1016/j.cie.2020.107082
6. H. Yapici, N. Cetinkaya, A new meta-heuristic optimizer: Pathfinder algorithm, *Appl. Soft Comput.*, **78** (2019), 545–568. https://doi.org/10.1016/j.asoc.2019.03.012
7. S. H. S. Moosavi, V. K. Bardsiri, Poor and rich optimization algorithm: A new human-based and multi populations algorithm, *Eng. Appl. Artif. Intell.*, **86** (2019), 165–181. https://doi.org/10.1016/j.engappai.2019.08.025
8. A.W. Mohamed, A. A. Hadi, A. K. Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm, *Int. J. M*ach*. Learn. Cybern.*, **11** (2020), 1501–1529. https://doi. org/10.1007/s13042-019-01053-x
9. Y. Liu, R. Li, PSA: A photon search algorithm. *J. Inf. Process. Syst.*, **16** (2020), 478–493. https://doi.org/10.3745/JIPS.04.0168
10. L. Pflug, N. Bernhardt, M. Grieshammer, CSG: A new stochastic gradient method for the efficient solution of structural optimization problems with infinitely many states, *Struct. Multidisc. Optim.*, **61** (2020), 2595–2611. https://doi.org/10.1007/s00158-020-02571-x
11. Y. Yang, Y. Gao, S. Tan, An opposition learning and spiral modelling based arithmetic optimization algorithm for global continuous optimization problems, *Eng. Appl. Artif. Intell.*, **113** (2022), 104981. https://doi.org/10.1016/j.engappai.2022.104981
12. A. T. Khan, X. Cao, S. Li, Dual beetle antennae search system for optimal planning and robust control of 5-link biped robots, *J. Comput. Sci.*, **60** (2022), 1877–7503. https://doi.org/10.1016/j.jocs.2022.101556
13. W. L. Liu, Y. J. Gong, W. N. Chen, Z. Liu, H. Wang, J. Zhang, Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach, *IEEE Trans. Intell. Transp.*, **21** (2020), 5094–5109. https://doi.org/10.1109/TITS.2019.2948596
14. F. Zhao, R. Ma, L. Wang, A self-learning discrete jaya algorithm for multi-objective energy-efficient distributed no-idle flow-shop scheduling problem in heterogeneous factory system, *IEEE Trans. Cybern.*, **247** (2021), 1–12. https://doi.org/ 10.1109/TCYB.2021.3086181
15. L. Pflug, N. Bernhardt, M. Grieshammer, CSG: A new stochastic gradient method for the efficient solution of structural optimization problems with infinitely many states, *Struct. Multidisc. Optim.*, **61** (2020), 2595–2611. https://doi.org/10.1007/s00158-020-02571-x
16. G. Dhiman, M. Garg, MoSSE: A novel hybrid multi-objective meta-heuristic algorithm for engineering design problems, *Soft Comput.*, **24** (2020), 18379–18398. https://doi.org/10.1007/s00500-020-05046-9
17. F. Zhao, S. Di, J. Cao, J. Tang, Jonrinaldi, A Novel Cooperative Multi-Stage Hyper-Heuristic for Combination Optimization Problems, *Complex Syst. Model. Simul.*, **1** (2021), 91-108. https://doi.org/10.23919/CSMS.2021.0010
18. Y. Yang, C. Qian, H. Li, An efficient DBSCAN optimized by arithmetic optimization algorithm with opposition-based learning, *J. Supercomput.*, **440** (2022). https://doi.org/10.1007/s11227-022-04634-w

19. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
20. S. Chakraborty, A. K. Saha, S. Sharma, S. Mirjalili, R. Chakraborty, A novel enhanced whale optimization algorithm for global optimization, *Comput. Ind. Eng.*, **153** (2021), 107086. https://doi.org/10.1016 /j.cie.2020.107086
21. Q. Jin, Z. Xu, W. Cai, An improved whale optimization algorithm with random evolution and special reinforcement dual-operation strategy collaboration, *Symmetry*, **13** (2021), 238–256. https:// doi. org/10.3390/sym13020238
22. J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, C. Li, Multi-strategy boosted mutative whale-inspired optimization approaches, *Appl. Math. Modell.*, **73** (2019), 109–123. https://doi.org/10.1016/j.apm.2019.03.046
23. Y. Yang, Z. Tao, C. Qian, A hybrid robust system considering outliers for electric load series forecasting, *Appl. Intell.*, **52** (2022), 1630–1652. https://doi.org/10.1007/s10489-021-02473-5
24. Q. Fan, Z. Chen, W. Zhang, ESSAWOA: Enhanced whale optimization algorithm integrated with salp swarm algorithm for global optimization, *Eng. Comput.*, **20** (2020). https://doi.org/10.1007/s00366-020-01189-3
25. M. Nguyen, T. Tran, T. Nguyen, G. Nguyen, Hybridization of galactic swarm and evolution whale optimization for global search problem, *IEEE Access*, **8** (2020), 74991–75010. https://doi.org/10.1109/ACCESS.2020.2988717
26. M. Abdel-Basset, V. Chang, R. Mohamed, HSMA_WOA: A hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images, *Appl. Soft Comput.*, **95** (2020), 1568–4946. https://doi.org/10.1016/ j.asoc.2020.106642
27. J. Zhang, L. Hong, Q. Liu, An improved whale optimization algorithm for the traveling salesman problem, *Symmetry*, **13** (2020), 48–56. https://doi.org/10.3390/sym13010048
28. M. Wang, H. Chen, Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis, *Appl. Soft Comput.*, **88** (2020), 105946. https://doi.org/10.1016/j.asoc.2019.105946
29. V. Hemmelmayr, K. F. Doerner, R. F. Hartl, S. Rath, A heuristic solution method for node routing based solid waste collection problems, *J. Heuristics*, **19** (2013), 129–156. https://doi.org/10.1177/0734242X18801186
30. F. Kılıç, Y. Kaya, S. Yildirim, A novel multi population based particle swarm optimization for feature selection, *Knowl.-Based. Syst.*, **219** (2021), 126–140. https://doi.org/10.1016/j.knosys.2021.106894
31. H. R. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, in *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, **1** (2005), 695–701. https://doi.org/10.1109/CIMCA.2005.1631345
32. S. Dhargupta, M. Ghosh, S. Mirjalili, R. Sarkar, Selective opposition based grey wolf optimization, *Expert. Syst. Appl.*, **151** (2020), 113389. https://doi.org/10.1016/j.eswa.2020.113389
33. M. H. Qais, H. M. Hasanien, S. Alghuwainem, Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators, *Appl. Soft Comput.*, **86** (2020), 105937. https://doi.org/10.1016/j.asoc.2019.105937
34. S. P. Adam, S. N. Alexandropoulos, P. M. Pardalos, M. N. Vrahatis, No free lunch theorem: A review, *Optim. Appl.*, **145** (2019), 57–82. https://doi.org/10.1007/978-3-030-12767-1_5

35. Q. Fan, Z. Chen, Z. Li, A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems, *Eng. Comput.*, **34** (2020), 1851–1878. https://doi.org/10.1007/s00366-019-00917-8
36. S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, *Knowl.-Based. Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
37. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
38. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based. Syst.*, **89** (2015), 228–249. https://doi.org/10.1016 /j.knosys.2015.07.006
39. H. Chen, W. Li, X. Yang, A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems, *Expert. Syst. Appl.*, **158** (2020), 23–46. https://doi.org/10.1016/j.eswa.2020.113612
40. W. Qiao, Z. Yang, Z. Kang, Short-term natural gas consumption prediction based on Volterra adaptive filter and improved whale optimization algorithm, *Eng. Appl. Artif. Intell.*, **87** (2020), 103323. https://doi.org/10.1016/j.engappai.2019.103323
41. T. Tiwari, M. Saraswat, A new firefly algorithm-based superpixel clustering method for vehicle segmentation, *Appl. Soft Comput.*, **278** (2022), 1–14. https://doi.org/10.1007/s00500-022-07206-5
42. A. Darwish, Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications, *Future Comput. Inf. J.*, **3** (2018), 231–246, https://doi.org/10.1016/j.fcij.2018.06.001
43. S. Zhou, A quantum image encryption method based on DNACNot, *IEEE Access*, **8** (2020), 178336–178344. https://doi.org/10.1109/ACCESS.2020.3027964
44. H. Hong, L. Wang, H. Ahmad, J. Li, Y. Yang, C. Wu, Construction of DNA codes by using algebraic number theory, *Finite Fields Appl.*, **37** (2016), 328–343. https://doi.org/10.1016/j.ffa.2015.10.008
45. X. Wang, W. Xue, J. An, Image encryption algorithm based on LDCML and DNA coding sequence, *Multimedia Tools Appl.*, **80** (2021), 591–614. https://doi.org/10.1007/s11042-020-09688-7
46. W. Song, K. Cai, M. Zhang, C. Yuen, Codes with run-length and GC-content constraints for DNA-based data storage, *IEEE Commun. Lett.*, **22** (2018), 2004–2007. https://doi.org/10.1109/LCOMM.2018.2866566
47. T. T. Nguyen, K. Cai, K. A. S. Immink, Constrained coding with error control for DNA-based data storage, in *2020 IEEE International Symposium on Information Theory (ISIT)*, (2020), 694–699. https://doi.org/10.1109/ISIT44 484.2020.9174438
48. Y. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, C. L. Poh, Thermodynamically stable DNA code design using a similarity significance model, in *2020 IEEE International Symposium on Information Theory (ISIT)*, (2020), 21–26. https://doi.org/10.1109/ISIT44484.2020.9174468
49. B. Cao, X. Li, X. Zhang, B. Wang, X. Wei, Designing uncorrelated address constrain for DNA storage by DMVO algorithm, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **11** (2020), 866–877. https://doi.org/10.1109/ TCBB.2020.3011582
50. D. Limbachiya, M. K. Gupta, V. Aggarwal, Family of constrained codes for archival DNA data storage, *IEEE Commun. Lett.*, **22** (2018), 1–10. https://doi.org/10.1109/LCOMM.2018.2861867
51. B. Cao, Q. Zhang, B. Wu, Q. Wang, X. Zhang, X. Wei, Minimum free energy coding for DNA storage, *IEEE Trans. Nanobiosci.*, **20** (2021), 212–222. https://doi.org/10.1109 /TNB.2021.3056351

52. M. Levy, E. Yaakobi, Mutually uncorrelated codes for DNA storage, *IEEE Trans. Inf. Theory*, **65** (2018), 3671–3691. https://doi.org/10.1109/TIT.2018.2873138