Improved Particle Swarm Based on Elastic Collision for DNA Coding Optimization Design

Donglin Zhu¹,², Zuwei Huang², Linpeng Xie¹, Changjun Zhou¹∗
¹College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China
²School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

Corresponding author: Changjun Zhou (e-mail: zhouchangjun@zjnu.edu.cn).

ABSTRACT In DNA computing, the design of DNA coding sequences is an important factor affecting the reliability of DNA computing. In different DNA sequence designs, suitable constraints should be selected and the sequence design should be rationalized according to these constraints. In this paper, an improved particle swarm optimization algorithm based on elastic collision strategy (EC-PSO) is used to optimize the design of DNA sequences by using an adaptation function that satisfies multiple constraints. EC-PSO uses the idea of elastic collision to improve the optimal and worst positions within the population, introduces the flight means of the sparrow search algorithm (SSA) to enhance the search capability of the algorithm and increase the diversity of the population; then introduces the harmony search algorithm to the population is then fine-tuned to improve the quality of the solution. The effectiveness of the algorithm was verified by comparing it with the other six algorithms in eight test functions. Finally, the sequence designed was more reasonable in the DNA optimal design experiment.

INDEX TERMS DNA computing, DNA coding, Improved particle swarm of elastic collisions, Sparrow search algorithm, Harmony search algorithm

I. INTRODUCTION

With the rapid development of science and technology, the data generated by human beings every day is increasing exponentially, so research and development of high-performance computing is urgent. DNA computing is a high-performance computing technology based on biological DNA molecules. Compared with traditional electronic computers, it has advantages of less resource consumption, high computing efficiency and large storage capacity, and has better application scenarios in the era of big data. In 1994, Professor Adleman [1] proved the feasibility of biological computing with DNA molecule as the carrier, and pioneered DNA computing. Subsequently, DNA computing has also been applied in various fields, such as DNA nanotechnology [2], DNA image encryption [3], etc.

DNA coding is the core step of DNA computing. A good DNA coding sequence determines the reliability of DNA computing. So how to design high quality DNA coding sequence is an important problem in the field of DNA computing. The design of DNA coding sequence is generally subject to multiple constraints to meet the specified quality requirements, such as continuity, similarity, and H-measure. Therefore, DNA coding design can also be regarded as an optimization problem with multiple objective functions [4].

Metaheuristic algorithm is a mathematical model of solving optimization problems by inference and induction inspired by natural laws. Genetic algorithm [5] is the first known meta-heuristic algorithm, it shows excellent optimization performance of metaheuristic algorithm in path planning [6], image processing [7], shop scheduling [8] and other problems. Subsequently, particle swarm algorithm [9], tabu search algorithm [10], ant colony algorithm [11], artificial bee colony algorithm [12] and other meta-heuristic algorithms have been studied. They all play an important role in optimization.

In order to obtain high quality DNA sequences, meta-heuristic algorithms have been widely used in DNA coding sequence optimization in recent years. Xiao et al. [13] combined particle swarm optimization algorithm with chaotic search algorithm and proposed a new quantum chaotic group evolution algorithm to solve the DNA sequence optimization problem. Ibrahim et al. [14] proposed a binary particle swarm optimization algorithm to design DNA coding sequences. Xiao et al. [15] used a multi-swarm particle swarm...
optimization algorithm based on time-varying acceleration coefficient to optimize DNA coding. Yang et al. [16] introduced a niche crowding strategy into the invasive weed optimization algorithm and algorithm to solve the problem of DNA sequence design. Liu et al. [17] combined bat algorithm and particle swarm optimization algorithm with a hybrid strategy, and used fast non-dominated sequencing to calculate and sort the fitness of DNA coding, achieving good optimization results. Bano et al. [18] proposed a multi-objective meme generalized differential evolution algorithm to design reliable DNA sequences based on reverse learning and local search strategies. Yao et al. [19] proposed an activity-based bacterial foraging algorithm by considering the influence of bacterial activity on foraging. The quality of DNA coding sequence was improved by introducing bacterial activity regulation mechanism and competitive exclusion mechanism. Zhou et al. [20] proposed an improved Bloch ball ant colony algorithm, which uses The Bloch system to initialize the population and then uses the ant colony algorithm to search for the optimal DNA code. Li et al. [21] proposed the closed constraint and paired sequence constraint on the original constraint, and obtained stable and reliable DNA sequence in combination with the improved chaotic whale algorithm. A great deal of work by researchers shows that meta-heuristic algorithms have obvious advantages in solving DNA coding problems. However, when the scale of DNA coding problem is enlarged, the optimization time of the algorithm is greatly increased and the efficiency is not high. The quality of DNA coding is also difficult to be guaranteed due to the strong randomness of metaheuristic algorithm, which is easy to fall into local optimization. Therefore, it is necessary to further improve the relevant algorithms to better solve the DNA coding problem.

Particle swarm optimization (PSO) is a classical meta-heuristic algorithm, it has good optimization ability and has few parameters and is easy to realize, but it also has some defects such as large randomness and easy local convergence. To solve these problems, scholars have made many improvements. Recently, Zhang et al. [22] proposed a multi-objective particle swarm optimization algorithm with control parameters. An improved position updating formula was used to replace the inertia weight and acceleration coefficient in traditional PSO, and an adaptive disturbance updating strategy was added to improve the universality and optimization ability of the algorithm. Zhang et al. [23] uses fuzzy clustering to guide population initialization, and adds local operators based on feature importance to strengthen local search ability, and finally obtains an improved particle swarm optimization algorithm with good optimization ability. Although the above improvements have effectively enhanced the local search capability of the particle swarm algorithm, it is still lacking in the global search.

In order to balance the local search and global search capability of the algorithm, this paper proposes an improved particle swarm algorithm based on elastic collision. And the following work has been carried out in this paper: Compared with Sparrow Search Algorithm (SSA) [24], PSO, Grey Wolf Optimizer (GWO) [25], Teaching learning based optimization (TLBO) [26], Manta Ray Foraging Optimization (MRFO) [27] and whale Optimization Algorithm (WOA)[28] on 8 standard test functions, the results show that EC-PSO has better optimization ability. Through the designed DNA sequence, it can be seen that EC-PSO has better rationality and higher quality than the algorithm proposed before. The main contributions of this paper are as follows:

1) An elastic collision strategy is proposed to improve the updating mode of individual position and enhance the local search capability of the algorithm.
2) In the iterative process, the finder-phase update strategy of sparrow search algorithm is introduced to enhance the global search capability of the algorithm.
3) harmony search algorithm was used to screen the individuals in the population, and the individuals with lower fitness value were eliminated, and the solution with higher quality was obtained.

In the rest of this paper, the second section introduces the constraints of DNA coding sequences. The third section introduces the improved strategy of improved particle swarm optimization algorithm based on elastic collision. The fourth section introduces the comparison and analysis of EC-PSO and other optimization algorithms in function optimization. The fifth section introduces the comparison and analysis of EC-PSO and other optimization algorithms in DNA coding sequence design. The sixth section summarizes the conclusion of this paper and points out the next step.

II. Constraints on DNA coding sequences

In 2004, Garzon et al. [29] discussed the problem of coding design for storing data in DNA sequences. DNA coding design should theoretically follow two major constraints, namely combinatory constraints and thermodynamic constraints. Common combinatory constraints include H-measure and similarity to prevent irregular hybridization of DNA molecules. And continuity and hairpin structures to prevent DNA molecules from forming the wrong structure during hybridization. Common thermodynamic constraints include GC content, chain temperature and free energy; These constraints allow DNA molecules to maintain consistent biochemical properties when hybridized. Next, the constraints used in this article are introduced.

In the following formula to introduce constraints, \( x \) and \( y \) respectively refer to the two DNA coding sequences in the DNA sequence set \( L \), \( m \) is the number of DNA sequences, and \( n \) is the number of bases in the given DNA sequence. Other functions involved are as follows.

\[
T(i, T_{\text{value}}) = \begin{cases} 
    i & \text{if } i > T_{\text{value}} \\
    0 & \text{otherwise}
\end{cases}
\] (1)
\[ cb(x, y) = \begin{cases} 1 & x, y \text{ complementary} \\ 0 & \text{otherwise} \end{cases} \quad (2) \]
\[ eq(a, b) = \begin{cases} 1 & a = b \\ 0 & \text{otherwise} \end{cases} \quad (3) \]

In formula (1), \( T(i, \text{value}) \) is a threshold function that returns \( i \) if parameter \( i \) is greater than the specified threshold \( \text{value} \), and 0 otherwise. In formula (2), \( cb(x, y) \) is the function that determines whether the bases \( x \) and \( y \) are complementary. In formula (3), \( eq(a, b) \) is the function that determines whether bases \( a \) and \( b \) are equal.

### A. Continuity

Continuity refers to the number of consecutive occurrences of the same base in a sequence of DNA code. If the frequency of occurrence is greater than the set threshold, the DNA molecular structure will become very unstable due to the hydrogen bonding between the bases, and it is easy to distort or fold during the reaction, resulting in secondary structure. DNA molecules with secondary structure cannot be used in DNA calculation, which will affect the reliability of calculation. So the continuity of the DNA coding sequence should be as small as possible. Continuity constraints can be expressed by a mathematical model as follows:

\[ f_{\text{continuity}}(L) = \sum_{p=1}^{m} \text{Continuity}(L_p) \quad (4) \]

\[ \text{Continuity}(x) = \sum_{i=1}^{n-CY} T(\text{con}(x, i), CY) \quad (5) \]

\[ \text{con}(x, i) = \begin{cases} c & \text{if } \exists c \text{ s.t. } x_i \neq b, x_{i+c+1} \neq b \\ x_{i+j} = b \text{ for } 1 < j \leq c \quad (6) \\ 0 & \text{otherwise} \end{cases} \]

In formula (4)-(6), \( CY \) is the set continuity threshold. In different DNA sequence lengths, it is generally required that \( CY \) should not be greater than 2 to reduce the probability of secondary structure of DNA sequence in the reaction process \( \text{con}(x, i) \) returns the number of consecutive bases, \( b \) being one of A, T, C, or G.

### B. Hairpin

Hairpin structure refers to the reverse folding of DNA molecule itself, with some bases close to each other and some folded bases complementing each other to form a secondary structure. It is called hairpin structure because its shape resembles hairpin. In DNA coding sequence, lowering hairpin structure value can reduce the self-reaction of DNA molecule, thus avoiding the emergence of secondary structure. Hairpin structure includes hairpin stem and hairpin ring, set the minimum stem length as \( P_{\text{min}} \), set the minimum ring length as \( R_{\text{min}} \), hairpin structure ring length as \( r \), stem length as \( q \), the calculation formula of hairpin structure is as follows:

\[ f_{\text{hairpin}}(L) = \sum_{p=1}^{m} \text{Hairpin}(L_p) \quad (7) \]

\[ \text{Hairpin}(x) = \sum_{q=Q_{\text{min}}}^{n-R_{\text{min}}} \sum_{r=R_{\text{min}}}^{n-2q} \sum_{i=1}^{q} T(\sum_{j=1}^{p} cb(x_{i+j}, x_{n-i}), \frac{P L_{\text{qir}}}{2}) \quad (8) \]

In formula (8), \( PL_{\text{qir}}=\min(p + i, l - r - i - p) \)

### C. Similarity

Similarity refers to the degree to which the base structures of two DNA molecules are similar. Hamming distance refers to the number of different bases in the corresponding positions of two DNA sequences, while similarity is a constraint condition based on Hamming distance. On the basis of hamming distance, the similarity constraint also considers the shift. Transposition refers to the fact that the hamming distance between DNA sequence \( x \) and \( y \) is very large, but the hamming distance becomes very small after the complementary strand of \( y \) is moved one bit to the right, so that the complementary strand of \( x \) and \( y \) is also prone to non-specific hybridization. Therefore, the smaller the similarity between DNA sequences, the lower the probability of non-specific hybridization between them. The calculation formula of similarity is as follows:

\[ f_{\text{similarity}}(L) = \sum_{p=1}^{m} \sum_{q=1}^{n} \text{Similarity}(L_p, L_q) \quad (9) \]

\[ \text{Similarity}(x, y) = \text{Max}_{\gamma,t}(s_{\text{dis}}(x, F\text{Shift}(y(-)^{\gamma}y, t)) + s_{\text{cont}}(x, F\text{Shift}(y(-)^{\gamma}y, t))) \quad (10) \]

\[ s_{\text{dis}}(x, y) = T(\sum_{i=1}^{n} eq(x_i, y_i), DP \times n) \quad (11) \]

\[ s_{\text{cont}}(x, y) = T(\text{subeb}(x, y, i), CP) \quad (12) \]

In formula (9)-(12), \( L_p, L_q \) are two coding sequences in DNA sequence set \( L \) and the similarity value is divided into two parts for calculation. One is discontinuous similarity, and the other describes the largest continuous common subset. \( y(-)^{\gamma}y \) represents the sequence of two pieces of \( y \) and the gap between the two pieces of \( y \) is \( g \). \( t \) represents the number of digits \( x \) moves to the right. \( \text{subeb}(x, y, i) \) returns the number of bases \( x \) and \( y \) are continuously equal starting at position \( i \), \( DP \in [0,1] \), \( CP \in [1, n] \) is a positive integer.

### D. H-measure

H-measure also calculates the possibility of unexpected base pairing between two DNA sequences based on hamming distance, and avoids mismatched hybridization between \( x \) and \( y \) by calculating the number of complementary bases between DNA sequences \( x \) and \( y \). At the same time, the sequence shift is also considered in the calculation process. The smaller the value of H-measure is, the lower the possibility of cross-linking with complementary strands of the same group of
DNA molecules. The calculation formula of H-Measure is as follows:

\[
f_{H-measure}(L) = \sum_{p=1}^{m} \sum_{v=1, p \neq v}^{m} H - measure(L_p, L_v) \tag{13}
\]

\[
H - measure(x, y) = \max_{[\theta]} (h_{dis(x, y)} + h_{cont}(x, y)) = \max_{[\theta]} (h_{cont}(x, y)) \tag{14}
\]

\[
h_{dis}(x, y) = T \sum c_b(x_i, y_i, DP \times n) \tag{15}
\]

\[
h_{cont}(x, y) = T \sum (subcb(x, y, i, CP) \tag{16}
\]

In formula (13)-(16), \(L_p, L_v\) are two reverse parallel DNA sequences. h-measure calculation can be divided into continuous calculation and discontinue calculation. And \(subcb(x, y, i)\) represents the number of consecutive complementary pairs of bases \(x\) and \(y\) starting from position \(i\).

E. GC content

GC content refers to the percentage of the number of guanine (G) and cytosine (C) in the DNA sequence to the total number of DNA bases. GC content is very important to keep the chemical properties of DNA sequence stable. In DNA computing, GC content of DNA coding sequence should be as consistent as possible, generally around 50%. Because \(G \equiv C\) contains three hydrogen bonds, and \(A \equiv T\) contains two hydrogen bonds, GC content also indirectly affects melting temperature.

\[
GC(x) = 100 \sum_{i=1}^{n} GC(x_i)/n \tag{17}
\]

\[
GC(a) = \begin{cases} 1, & a = G \text{ or } a = C \\ 0, & a = A \text{ or } a = T \end{cases} \tag{18}
\]

F. Melting temperature

Melting temperature (Tm) is the temperature at which 50% of the double stranded structure of DNA double stranded molecule is opened into a single strand under heating conditions. Tm is an important factor affecting the reaction efficiency of DNA molecules. In DNA computing, it is required that the DNA coding sequence should have the same melting temperature as far as possible, so as to better control the reaction between DNA molecules and effectively reduce the probability of non-specific hybridization. According to the nearest neighbor thermodynamics model \([30]\), the calculation formula of Tm is as follows:

\[
f_{Tm}(L) = \frac{\Delta H^\circ}{\Delta S^\circ + R \ln([C_T]/4)} = 273.15 \tag{19}
\]

In formula (19), \(\Delta H^\circ\) is the total enthalpy of adjacent bases, \(\Delta S^\circ\) is the total entropy of adjacent bases. \(R\) is the gas constant (1.987 cal/kmol), \(C_T\) is the concentration of DNA.

III. Introduction to EC-PSO algorithm

In standard particle swarm, individuals need to update their positions according to the current population information, so strengthening the communication of position information between particles in the population can facilitate optimization. It is worth noting that in the search for optimization in complex environment, the range of change of the best and worst positions in the population is small, or even unchanged after multiple searches, which will limit the search ability of the algorithm and make the subsequent search meaningless. Scientific researchers have found such problems and also carried out partial update for the optimal location. However, the traditional update strategy will evaluate the individual after searching for several times, which increases the computational complexity of the algorithm, so the effect is not comprehensive enough.

A. Elastic Collision strategy

Elastic collision comes from the knowledge of mechanics in physics. When ball A hits another static ball B at speed V, the speed and direction will change. Assuming no energy is lost during the collision, and a initially moves in A positive direction, there are three situations after the collision.

1. If the mass of A is greater than that of B, the two balls move in the positive direction after the collision.

2. If the mass of A is less than that of B, A moves in the opposite direction and B moves in the positive direction after the collision.

3. If the mass of A is equal to the mass of B, A is at rest after the collision, and B moves to the right with A velocity equal to the original velocity V.

In the design of this paper, the worst individual in the population is endowed with speed to impact the best individual, and the fitness value of the individual is equivalent to the mass of the object. The fitness value of the position after the collision is compared to select the best and worst individual respectively. The specific idea is shown in Figure 1.

As shown in Figure 1, the fitness value of the worst individual is \(m_1\), and the fitness value of the best individual is \(m_2\). The worst individual of the worst individual \(m_1\) is assigned to speed, and the velocity can be calculated as follows:

\[
v_1 = Z \cdot e^{-x_{worst} - x_{ran}} \tag{20}
\]

\[
Z = \frac{1}{2} \left(1 - 2 \times 1^3 + 1\right) \tag{21}
\]

In formula (20)-(21), \(x_{worst}\) represents the worst individual position, \(x_{ran}\) represents the position of randomly selected individuals. The difference between the two is used to perceive the location information of the population and master certain group information. \(Z\) is a non-linear parameter factor used to improve speed adaptability. where \(T\) is the maximum number of iterations and \(t\) is the current number of iterations is considered as the independent variable., and the velocity \(v_1\) is seen as the dependent variable. The velocity change diagram is shown in Figure 2.
It can be seen that with the increase of the number of iterations, the speed gradually increases and the transformation is faster, which is conducive to the search ability in the middle and late period and reduces the probability of falling into local optimal.

\[ \begin{align*}
\text{worst} & \quad v_1 \quad \text{best} \\
\text{worst}' & \quad v_1' \quad \text{best}'
\end{align*} \]

\[ \begin{align*}
X_{\text{worst}}' &= X_{\text{worst}} + v_1' \quad (25) \\
X_{\text{best}}' &= X_{\text{best}} + v_2' \quad (26)
\end{align*} \]

Thus, the velocity of the worst individual \( v_1' \) and the velocity of the best individual \( v_2' \) are calculated:

\[ \begin{align*}
v_1' &= \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \quad (23) \\
v_2' &= \frac{2m_1}{m_1 + m_2} \cdot v_2 \quad (24)
\end{align*} \]

According to the principle of elastic collision and momentum conservation law, it can be obtained:

\[ \begin{align*}
m_1 v_1 &= m_1 v_1' + m_2 v_2' \\
\frac{1}{2} m_1 v_1^2 &= \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2
\end{align*} \quad (22) \]

Thus, the velocity of the worst individual \( v_1' \) and the velocity of the best individual \( v_2' \) are calculated:

\[ \begin{align*}
v_1' &= \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \\
v_2' &= \frac{2m_1}{m_1 + m_2} \cdot v_2
\end{align*} \]

So the final position update formula is:

\[ \begin{align*}
X_{\text{worst}}' &= X_{\text{worst}} + v_1' \\
X_{\text{best}}' &= X_{\text{best}} + v_2'
\end{align*} \]

\[ \begin{align*}
X_{\text{worst}}' &= X_{\text{worst}} + v_1' \\
X_{\text{best}}' &= X_{\text{best}} + v_2'
\end{align*} \]

\[ \begin{align*}
X_{\text{worst}}' &= X_{\text{worst}} + v_1' \\
X_{\text{best}}' &= X_{\text{best}} + v_2'
\end{align*} \]

According to the principle of elastic collision and momentum conservation law, it can be obtained:

\[ \begin{align*}
m_1 v_1 &= m_1 v_1' + m_2 v_2' \\
\frac{1}{2} m_1 v_1^2 &= \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2
\end{align*} \quad (22) \]

Thus, the velocity of the worst individual \( v_1' \) and the velocity of the best individual \( v_2' \) are calculated:

\[ \begin{align*}
v_1' &= \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \\
v_2' &= \frac{2m_1}{m_1 + m_2} \cdot v_2
\end{align*} \]

According to the principle of elastic collision and momentum conservation law, it can be obtained:

\[ \begin{align*}
m_1 v_1 &= m_1 v_1' + m_2 v_2' \\
\frac{1}{2} m_1 v_1^2 &= \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2
\end{align*} \quad (22) \]

Thus, the velocity of the worst individual \( v_1' \) and the velocity of the best individual \( v_2' \) are calculated:

\[ \begin{align*}
v_1' &= \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \\
v_2' &= \frac{2m_1}{m_1 + m_2} \cdot v_2
\end{align*} \]

According to the principle of elastic collision and momentum conservation law, it can be obtained:

\[ \begin{align*}
m_1 v_1 &= m_1 v_1' + m_2 v_2' \\
\frac{1}{2} m_1 v_1^2 &= \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2
\end{align*} \quad (22) \]

Thus, the velocity of the worst individual \( v_1' \) and the velocity of the best individual \( v_2' \) are calculated:

\[ \begin{align*}
v_1' &= \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \\
v_2' &= \frac{2m_1}{m_1 + m_2} \cdot v_2
\end{align*} \]
In Formula (27), \( \text{Maxitem} \) represents the maximum number of iterations of the algorithm. \( \alpha \) is a uniform number conforming to \((0, 1)\); \( Q \) is a random number that follows the standard normal distribution; \( L \) is a \( 1 \times d \) matrix where each entry is 1; Alarm value \( R_2 \in [0, 1] \) and safety value \( ST \in [0.5, 1] \). If a sparrow in the population finds danger, it will send out an alarm signal. When the alarm value is greater than the safe value, the finder will take the population to another safer area to forage.

In this paper, the finder-stage optimization strategy of SSA is used to update the particle position, which makes the subsequent optimization more reliable. Firstly, some individuals are randomly selected from the population for updating according to Equation (27), providing two individuals with large differences for the collision process, so that the next optimization range is more extensive. Some individuals are selected in a proportional manner. The proportion of individuals selected is set as 0.1. In this way, the selected individuals will not greatly increase the computational complexity of the algorithm. In addition, the diversity of the population is increased to some extent, and the optimization ability of the algorithm is improved.

### C. Harmony search algorithm

Harmony search algorithm (HS) [31] the main idea from the harmony memory of randomly generated, based on considering the harmony memory, random selection and the operation of the pitch adjustment strategies candidate solution vectors, then a candidate solution vector and harmony the worst memory solution vector fitness function value comparison, decide whether to update the harmony memory Banks. In this paper, HS is used to locally adjust the individuals of the current population and eliminate the worst solution in the current candidate solution, so as to prevent the updated worst solution from continuously affecting the search results, so as to extract a group of high-quality solutions.

### D. EC-PSO algorithm

strongly encouraged.) English units may be used as secondary In order to improve the optimization ability of particle swarm optimization and increase the diversity of its optimization methods, an improved particle swarm optimization algorithm based on elastic collision is proposed in this paper. The algorithm uses elastic collision to improve the best and worst individuals in the population without increasing the computational complexity of the algorithm. The finders update formula of the SSA is introduced to improve the global search ability of the algorithm. Finally, HS is used to update the candidate solutions and retain a set of high-quality solutions. The specific process is as follows:

Step 1. Initialize the population and parameters.

Step 2. Use the formula (25) to update the best and worst individuals.

Step 3. Use the formula (27) to update the corresponding position.

Step 4. Update the corresponding position and speed.

Step 5. Fine-tuning the population with HS.

Step 6. Generate the corresponding optimal position and optimal solution.

Step 7. If iteration ends, output the optimal position and solution; otherwise, return to step 2.

### E. Time complexity analysis of algorithm

Time complexity is an important factor to measure the quality of the algorithm and determines the rationality and timeliness of the algorithm. Let the population number of EC-PSO be \( N \), the maximum iteration times be \( T \), the dimension of the problem be \( D \), and the randomly selected ratio be \( R_1 \). Therefore, the time complexity of EC-PSO is analyzed as follows:

From the macro point of view, the time of swarm intelligence optimization algorithm is \( O(N \times T \times D) \), and EC-PSO is the same. Although EC-PSO added some strategies in the optimization process, it did not change the structure of the algorithm and increase the number of cycles, so its time complexity was still \( O(N \times T \times D) \).

From the microscopic point of view, it is assumed that the calculation time for introducing elastic collision is \( T_1 \), that for adding global search is \( T_2 \), and that for introducing harmony search is \( T_3 \). Other parts of the calculation scale is small and can be ignored. As can be seen from the algorithm flow chart, \( O(T \times 2) \) is added in the phase of updating the best and worst locations, \( O(T \times R_1 \times N \times D) \) is added in the global search phase, and \( O(T \times N \times D) \) is added in the introduction of HS. Therefore, compared with PSO, the time complexity of EC-PSO is increased by \( O(T(2 + N \times D \times (R_1 + 1))) \). But the order of magnitude is not improved, and the optimization efficiency and accuracy of the algorithm can be effectively improved, so the increased time complexity is significant and worthwhile.

### F. DNA sequence design process based on EC-PSO

In this paper, an improved particle swarm optimization algorithm for elastic collisions is proposed for DNA sequence optimization, and the fitness function is a combination of continuity, hairpin and Hamming constraints. When the DNA sequence meets this particular condition, the fine-tuned individual will be retained for subsequent operations, otherwise it will be eliminated. The aim is to prevent secondary structures from developing among individuals in the population and to reduce similarity to some extent. The magnitude of each fine tuning of harmonic search is determined by the fine tuning probability. When the number of iterations reaches the common multiple of 100, the population is re-initialized to prevent the algorithm from entering the design mistakes, so as to get rid of the influence.
of the initialization stage. The specific flow chart is shown in Figure 3.

![Flow chart of EC-PSO DNA sequence design](image)

V. Algorithm performance test and analysis

In the performance test experiment, eight standard test functions were used to verify the validity of EC-PSO. In order to make the test more reasonable and comprehensive, different function types were used in the experiment to test the optimization performance of the algorithm. The specific function information is shown in Table 1. F1-F4 is a simple single-peak function, F5-F7 is a complex multi-peak function, and F8 is a fixed-dimensional function. In addition, PSO, GWO, SSA, MRFO, TLBO and WOA algorithms are compared with EC-PSO to verify the advanced nature of EC-PSO. In order to better show the optimization of each algorithm, each algorithm runs 30 times, the best value, worst value, median, mean, standard deviation, run time of each algorithm are calculated. In order to be fair, the population (pop) size and iteration (iter) times of each algorithm remain the same. The specific parameters of each algorithm are shown in Table 2. All experiments were run on a PC with CPU of Core i5-10200h, RAM of 16G, operating system of Windows 10, and compilation environment of MATLAB R2019a. The optimization results obtained in the experiment are shown in Table 3.

As can be seen from Table 3, EC-PSO can find the optimal value in F1-F4 function every time, while SSA and MRFO can find the optimal value in most cases, but there will be deviations occasionally, with poor stability in searching, and other algorithms perform worse. It can be seen that EC-PSO has excellent performance in single peak optimization. In terms of multi-peak function, EC-PSO is slightly different from SSA and MRFO, but it still outperforms other algorithms, proving that EC-PSO has good performance in multi-peak problem. Finally, the performance of EC-PSO is almost the
same as that of MRFO in the fixed dimension function, which is better than other algorithms. In conclusion, multi-strategy supported EC-PSO algorithm has good optimization performance. In order to clearly see the convergence of each algorithm in the function, the average convergence effect diagram of each algorithm in each function is given, as shown in Figure 4. To show the difference of the algorithms, the horizontal coordinates in some functions are reduced so that the difference of the algorithms can be seen before the search.

TABLE 1. List of performance test functions

| F        | DIM | INTERVAL      | MIN |
|----------|-----|---------------|-----|
| $F_1(x) = \sum_{i=1}^{n} x_i^2$ | 30   | [-100,100]    | 0   |
| $F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} x_i$ | 30   | [-10,10]      | 0   |
| $F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | 30   | [-100,100]    | 0   |
| $F_4(x) = \max_i(|x_i|, 1 \leq i \leq n)$ | 30   | [-100,100]    | 0   |
| $F_5(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | 30   | [-5.12,5.12]  | 0   |
| $F_6(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$ | 30   | [-600,600]    | 0   |
| $F_7(x) = \frac{10}{\pi} \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[ 1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2 \right]$ | 30   | [-50,50]      | 0   |
| $\sum_{i=1}^{n} u(x_i, 10, 100, 4)$ | 30   | [-50,50]      | 0   |
| $y_i = 1 + \frac{x_i + 1}{4}$ | 30   | [-50,50]      | 0   |
| $\frac{1}{25} \sum_{i=1}^{25} [x_i - a_i]^2 + (x_2 - a_{25})^2$ | 2    | [-65.536,65.536] | 0.998 |

TABLE 2. Parameter table of each algorithm

| algorithms | PSO | GWO | TLBO | SSA | WOA | MRFO | EC-PSO |
|------------|-----|-----|------|-----|-----|------|--------|
| parameters | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 | Pop=100, Iter=500, C1=1.4, C2=1.4, W=0.928 |

TABLE 3. Test results of each algorithm

| F | index | PSO | GWO | TLBO | SSA | WOA | MRFO | EC-PSO |
|---|-------|-----|-----|------|-----|-----|------|--------|
| Mean | 7.3054E-11 | 9.9979E-41 | 2.9874E-85 | 0   | 1.0841E-95 | 0   | 0   |
| $F_1(x)$ Best | 1.356E-12 | 2.9272E-42 | 4.053E-86 | 0   | 3.1181E-104 | 0   | 0   |
| Std  | 1.538E-10 | 2.3113E-40 | 2.187E-85 | 0   | 4.1034E-95 | 0   | 0   |
| F   | index | PSO  | GWO  | TLBO  | SSA  | WOA  | MRFO | EC-PSO |
|-----|-------|------|------|-------|------|------|------|--------|
|     | Best  | Mean | Median | Best  | Mean | Median | Best  | Mean  |
| F_1 | 1.0043E-01 | 1.0199E-01 | 2.5276E-01 | 1.2950E-00 | 1.2500E-00 | 3.5658E-00 | 8.2816E-01 | 6.8750E-01 |
| F_2 | 1.6315E-01 | 7.8686E-02 | 1.0753E-02 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |
| F_3 | 1.0043E-01 | 1.0199E-01 | 2.5276E-01 | 1.2950E-00 | 1.2500E-00 | 3.5658E-00 | 8.2816E-01 | 6.8750E-01 |
| F_4 | 1.6315E-01 | 7.8686E-02 | 1.0753E-02 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 | 0.0000E+00 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
| F       | index | PSO     | GWO     | TLBO    | SSA     | WOA     | MRFO    | EC-PSO  |
|---------|-------|---------|---------|---------|---------|---------|---------|---------|
| Std     | 3.436E-01 | 9.724E-01 | 0       | 3.362E+00 | 0       | 0       | 0       | 0       |
| Median  | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 |
| Worst   | 1.952E+00 | 2.982E+00 | 9.980E-01 | 1.267E+01 | 2.982E+00 | 9.980E-01 | 9.980E-01 | 9.980E-01 |
| Mean    | 1.135E+00 | 1.725E-01 | 9.980E-01 | 2.366E+00 | 1.262E+00 | 2.982E+00 | 9.980E-01 | 9.980E-01 |
| Run time| 4.687E-02 | 7.656E-01 | 2.078E+00 | 1.871E+00 | 7.796E+00 | 1.703E+00 | 2.093E+00 | 0       |

![Graphs](a)-(f)
Figure 4 shows that EC-PSO has a fast optimization speed and high accuracy. Especially, F1-F4 and F7 have obvious advantages in optimization effect and convergence speed. On the other hand, although EC-PSO does not achieve the best convergence effect on F5-F6 functions, it is only second to SSA and MRFO, and very close to MRFO on F8 functions. In summary, although EC-PSO has some differences with other algorithms in multi-peak problem, it is far superior to other algorithms in single-peak problem. Compared with the standard PSO algorithm, the improved EC-PSO optimization ability is greatly improved, thus proving the effectiveness and advance of the improved strategy proposed in this paper.

VI. DNA sequence optimization experiment based on EC-PSO

DNA sequence design needs to meet multiple constraints, so DNA sequence optimization can be regarded as an optimization problem with multiple objective functions. In order to verify the practicality of EC-PSO, EC-PSO was applied to DNA sequence optimization. In EC-PSO, a DNA sequence is a candidate solution, and the algorithm will find the best DNA sequence that meets the constraints. The experimental environment is the same as the algorithm performance test environment. Related parameters in DNA sequence optimization are: the minimum stem length and minimum ring length threshold of hairpin structure are both 6. The minimum continuity critical value is 2. In the continuous case, the critical value of similarity and H-measure is 6. In the case of discontinuity, the critical value of similarity and H-measure is 0.17. For codes with a length of 20 DNA Sequence Design Optimized. In addition, the concentration of DNA molecule is 10nM and the concentration of salt is 1mol/L. Table 3 shows the specific parameters of EC-PSO.

### Table 3. Parameters related to the algorithm.

| Symbol | significance | value |
|--------|--------------|-------|
| M | Maximum iteration | 3000 |
| pop | population capacity | 20 |
| pos | The length of the individual | 20 |
| F | Fine-tune probability | 0.5 |
| Wmin | Minimum dynamic weight factor | 0.4 |
| Wmax | Maximum dynamic weight factor | 0.9 |
| C1 | Initial individual learning factor | 2.5 |
| C1min | Minimum individual learning factor | 0.5 |
| C2 | Initial social learning factor | 2.5 |
| C2min | Minimum social learning factor | 0.5 |
| L | Hamming distance | 11 |

### Table 4. DNA sequence optimization results of each algorithm.

| DNA sequences | continuity | Hairpin | H-measure | Similarity | Tm | GC % |
|---------------|------------|---------|-----------|------------|----|------|
| NCIWO         | 9          | 0       | 55        | 46         | 66.99 | 50   |
| GTTCAATCGCTCTCGTAT | 0          | 0       | 57        | 52         | 64.26 | 50   |
| GCTACCTCTTCCACCATCT | 0          | 0       | 55        | 53         | 63.55 | 50   |
| GAATCAATGGGCAGTACAGAAG | 0         | 0       | 66        | 47         | 63.58 | 50   |
| TTTGTCGCCTATCTCTCGT | 0          | 0       | 65        | 52         | 64.44 | 50   |
| CCCATCTCCGTACTTACCTG | 0          | 0       | 56        | 56         | 62.30 | 50   |
| TTGACTCCGCTCTCTGCTA | 0          | 0       | 58        | 54         | 65.61 | 50   |
| average | 1.2857 | 0 | 58.8571 | 51.4286 | 64.3900 | 50 |

**TABLE 3.** DNA sequence optimization results of each algorithm.

**TABLE 4.** Parameters related to the algorithm.
In order to highlight the performance of EC-PSO in DNA optimization, NCIWO [32], IWO [33], CPSO [34] DMEA [35] and DNA sequences designed by NUPACK were selected for comparison. NUPACK is a software designed by Caltech academics that generates DNA sequence structures using code. In addition, the parameters of all algorithms are set according to the literature, and the constraint formula is consistent with this paper, which fully ensures the rationality and fairness of the experiment. In order to test the effectiveness of the algorithm in DNA sequence optimization, 7 DNA sequences containing 20 bases were generated in the experiment. In order to test the effectiveness of the algorithm in DNA sequence optimization, 7 DNA sequences containing 20 bases were generated in the experiment. The base arrangement of these sequences and the values of each constraint condition after the algorithm optimization are shown in Table 4.

| Sequence | Continuity | Hairpin | Average | 
|----------|------------|---------|---------|
| AAGGCCGAAGGAAGTAGAGG | 0.8571 | 0.8571 | 0.8643 |
| TTCTCACAGCTTCTTCTTC | 0.7143 | 0.7143 | 0.7143 |
| GACTAAGCACAACACAGCA | 0.5714 | 0.5714 | 0.5714 |
| GTGCACTACCCACACAAGGC | 0.5714 | 0.5714 | 0.5714 |
| GACTAACAACTCCTGCCGC | 0.5714 | 0.5714 | 0.5714 |
| CATAGAGGATGACAAAGGCTCC | 0.5714 | 0.5714 | 0.5714 |
| CATAGAGGATGACAAAGGCTCC | 0.5714 | 0.5714 | 0.5714 |

In DNA sequence design, the smaller the value of continuity and hairpin, the less likely the DNA is to produce secondary structures. The smaller the value of hairpin and similarity, the lower the probability that DNA molecule will hybridize incorrectly. The smaller the change of Tm and GC content, the better. According to Table 4, only the continuity and hairpin of EC-PSO are 0. The hairpin value of the last sequence in DMEA was 3. The continuity value of the first sequence in NCIWO is 9. The hairpin value of the first and third sequences in NUPACK are 3. The continuity and hairpin values of IWO and CPSO are not zero, so the optimization effect is poor. Therefore, EC-PSO can effectively avoid the secondary structure of DNA molecules. Figure 5 shows the average value of continuity and hairpin of each algorithm.

In EC-PSO, fine tuning operation of HS is adopted to remove individuals with low fitness value, which reduces similarity and H-measure value to a certain extent, which plays a very important role in DNA sequence optimization. Low similarity and H-measure values can effectively prevent DNA sequences from mismatching during reaction. It can also be seen from Figure 6 that the mean value of similarity and H-measure of EC-PSO are the minimum, indicating that DNA sequences optimized by EC-PSO can be hybridized more regularly during the reaction.
In DNA sequence, the optimal proportion of guanine (G) and cytosine (C) is 50%. In this experiment, under the condition of meeting other constraints, the GC content is optimized to 50%. In Table 4, only EC-PSO, NCIWO and CPSO achieve this goal. However, it can be seen from Figure 7 that the variance of melting temperature of NCIWO and CPSO is greater than EC-PSO, the change of melting temperature is greater. It is proved that the DNA sequences optimized by EC-PSO have better stability in biochemical reactions.

**REFERENCES**

[1] Adleman, Leonard M. "Molecular computation of solutions to combinatorial problems." Science 266.5187 (1994): 1021-1024.

[2] Wang, Wen, et al. "Complex wireframe DNA nanostructures from simple building blocks." Nature communications 10.1 (2019): 1-8.

[3] Liang, Zhongyue, et al. "Medical image encryption algorithm based on a new five-dimensional three-leaf chaotic system and genetic operation." Plos one 16.11 (2021): e0260014.

[4] Tang, Zhen, et al. "Solving 0-1 Integer Programming Problem Based on DNA Strand Displacement Reaction Network." ACS Synthetic Biology 10.9 (2021): 2318-2330.

[5] Goldberg, David E., and John Henry Holland. "Genetic algorithms and machine learning." (1988).

[6] Tao, WeiMin, and Mingjun Zhang. "A genetic algorithm–based area coverage approach for controlled drug delivery using microrobots." Nanomedicine: Nanotechnology, Biology and Medicine 1.1 (2005): 91-100.

[7] Munteanu, Cristian, and Agostinho Rosa. "Towards automatic image enhancement using genetic algorithms." Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512). Vol. 2. IEEE, 2000.

[8] Pasupathy, T., Chandrasekharan Rajendran, and R. K. Suresh. "A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs." The International Journal of Advanced Manufacturing Technology 27.7 (2006): 804-815.

[9] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. IEEE, 1995.

[10] Glover, Fred, and Manuel Laguna. "Tabu search." Handbook of combinatorial optimization. Springer, Boston, MA, 1998. 2093-2229.

[11] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." IEEE computational intelligence magazine 1.4 (2006): 28-39.

[12] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." Applied mathematics and computation 214.1 (2009): 108-132.

[13] Xiao, Jianhua, et al. "A hybrid quantum chaotic swarm evolutionary algorithm for DNA encoding." Computers & Mathematics with Applications 57.11-12 (2009): 1949-1958.

[14] Ibrahim, Zuwaire, et al. "A DNA sequence design for DNA computation based on binary vector evaluated particle swarm optimization strategy improved particle swarm optimization (EC-PSO) algorithm, the algorithm adopts the elastic collision theory to update the best and worst position, the introduction of SSA discoverer of update formula to improve the search of algorithm, and finally use HS to extract of populations, selecting better individuals. The effectiveness of the algorithm is verified by eight test functions. In DNA sequence design optimization, the sequences obtained by optimization are of higher quality, which can effectively meet the requirements of DNA computing. Due to the mutual constraints among constraints, we consider to further improve the algorithm and use non-dominated sequencing to screen more high-quality DNA sequences in future work.
D. Zhu et al.: EC-PSO for DNA Coding Optimization Design

optimization.” International Journal of Unconventional Computing 8.2 (2012): 119-137.

[15] Xiao, Jianhua, and Zhen Cheng. "A multi-swarm particle swarm optimization to solve DNA encoding in DNA computation.” Journal of Computational and Theoretical Nanoscience 10.5 (2013): 1129-1136.

[16] Yang, Gaijing, et al. "TWO algorithm based on niche crowding for DNA sequence design.” Interdisciplinary Sciences: Computational Life Sciences 9.3 (2017): 341-349.

[17] Liu, Kaiqiang, et al. "A BPSON algorithm applied to DNA codes design.” IEEE Access 7 (2019): 88811-88821.

[18] Bano, Shah, Maryam Bashir, and Irfan Younas. "A Many-Objective Memetic Generalized Differential Evolution Algorithm for DNA Sequence Design." IEEE Access 8 (2020): 222684-222699.

[19] Yao, Yao, et al. "Bacterial Foraging Algorithm Based on Activity of Bacteria for DNA Computing Sequence Design.” IEEE Access 9 (2020): 2110-2124.

[20] Zhou, Qiang, Xiao Wang, and Changjun Zhou. "DNA Design Based on Improved Ant Colony Optimization Algorithm With Bloch Sphere.” IEEE Access 9 (2021): 104513-104521.

[21] Li, Xue, et al. "Stable DNA Sequence Over Close-Ending and Pairing Sequences Constraint.” Frontiers in genetics 12 (2021).

[22] Yong, Zhang, et al. "Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters.” Journal of Building Engineering 32 (2020): 101505.

[23] Zhang, Yong, et al. "Clustering-guided particle swarm feature selection algorithm for high-dimensional imbalanced data with missing values.” IEEE Transactions on Evolutionary Computation (2021).

[24] Xue, Jiankai, and Bo Shen. "A novel swarm intelligence optimization approach: sparrow search algorithm.” Systems Science & Control Engineering 8.1 (2020): 22-34.

[25] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer.” Advances in engineering software 69 (2014): 46-61.

[26] Rao, R. Venkata, Vimal J. Savsani, and D. P. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems.” Computer-Aided Design 43.3 (2011): 303-315.

[27] Zhao, Weiguo, Zhenxing Zhang, and Lijing Wang. "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications.” Engineering Applications of Artificial Intelligence 87 (2020): 103300.

[28] Mirjalili, Seyedali, and Andrew Lewis. "The whale optimization algorithm.” Advances in engineering software 95 (2016): 51-67.

[29] Garzon, Max H., and Russell J. Deaton. "Codeword design and information encoding in DNA ensembles.” Natural Computing 3.3 (2004): 253-292.

[30] Watkins Jr, Norman E., and John SantaLucia Jr. "Nearest-neighbor thermodynamics of deoxyinosine pairs in DNA duplexes.” Nucleic acids research 33.19 (2005): 6258-6267.

[31] Geem, Zong Woo, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. "A new heuristic optimization algorithm: harmony search.” simulation 76.2 (2001): 60-68.

[32] Yang, Gaijing, et al. "TWO algorithm based on niche crowding for DNA sequence design.” Interdisciplinary Sciences: Computational Life Sciences 9.3 (2017): 341-349.

[33] Luo, D. F., and D. J. Luo. "The research of DNA coding sequences based on invasive weed optimization." Sci. Technol. Eng 13.13 (2013): 3545-3551.

[34] Liu, Yuanyuan, et al. "The optimization of DNA encoding based on chaotic optimization particle swarm algorithm.” Journal of Computational and Theoretical Nanoscience 13.1 (2016): 443-449.

[35] Xiao, J. H., et al. "A dynamic membrane evolutionary algorithm for solving DNA sequences design with minimum free energy.” MATCH Communications in Mathematical and in Computer Chemistry 70.3 (2013): 971-986.

FIRST A. Donglin Zhu, born in 1997, male. From Jiujiang, Jiangxi, Master's degree, major in computer technology, engaged in intelligent optimization algorithm, machine learning, image processing and other related research.

SECOND B. Zuwei Huang, born in 1998, male. From Ganzhou, Jiangxi, Master's degree, major in computer technology, engaged in intelligent optimization algorithm, machine learning, image processing and other related research.

THIRD C. Linpeng Xie, born in 1999, male. From Quzhou, Zhejiang, Currently a graduate student in Computer Science and Technology, School of Mathematics and Computer Science, Zhejiang Normal University. Mainly engaged in the research of DAN storage coding, DNA computing and other fields.
FOURTH D. Changjun Zhou, born in 1977 male, From Shangrao, Jiangxi, PhD, distinguished professor of "Shuanglong scholar" in the school of mathematics and computer science of Zhejiang Normal University. At present, he is mainly engaged in Intelligent Computing and pattern recognition in new computing models, biological computing theory and its application, software system development, etc.