



Improved clonal selection algorithm based on the directional update strategy

Chao Yang^{1,2,3} · Zijing Huang¹ · Bibo Jiang¹ · Menglian Zhu¹ · Aoran Luo¹ · Jianfeng He¹

Accepted: 15 May 2023 / Published online: 27 May 2023
© The Author(s) 2023

Abstract

At present, many cloning selection algorithms have been studied, and improvements have been made to the cloning, mutation and selection steps. However, there is a lack of research on the optimization of the updating operation steps. The clonal selection algorithm is traditionally updated through a random complement of antibodies, which is a blind and uncertain process. The added antibodies may gather near a local optimal solution, resulting in the need for more iterations to obtain the global optimal solution. To solve this problem, our improved algorithm introduces a crowding degree factor in the antibody updating stage to determine whether there is crowding between antibodies. By eliminating antibodies with high crowding potential and poor affinity, the improved algorithm guides the antibodies to update in the direction of the global optimal solution and ensures stable convergence with fewer iterations. Experimental results show that the overall performance of the improved algorithm is 1% higher than that of the clonal selection algorithm and 2.2% higher than that of the genetic algorithm, indicating that the improved algorithm is effective. The improved algorithm is also transplanted to other improved clonal selection algorithms, and the overall performance is improved by 0.97%, indicating that the improved algorithm can be a beneficial supplement to other improved clonal selection algorithms.

Keywords Clonal selection algorithm · Antibody update · Targeted update strategy · Antibody crowding · Antibody affinity

✉ Chao Yang
stevenyc@hubu.edu.cn

¹ College of Computer and Information Engineering, Hubei University, Youyi Avenue, Wuhan 430062, Hubei Province, China

² Hubei Provincial Education Information Engineering Technology Research Center, Wuhan 430062, Hubei Province, China

³ Hubei Province Project of Key Research Institute of Humanities and Social Sciences at Universities (Research Center of Information Management for Performance Evaluation), Wuhan 430062, Hubei Province, China

1 Introduction

The clonal selection algorithm (CSA) [1] was proposed based on the eponymous principle of biological systems that explains the immune response of B cells in organisms during the recognition of nonself antigens [2]. The clonal selection algorithm is a heuristic optimization algorithm that involves cloning and mutation of a single antibody to expand the search range. As a heuristic algorithm to solve complex problems with a high risk of mutation, the clonal selection algorithm converges quickly and has obvious advantages over other current mainstream intelligent optimization algorithms [3]. In recent years, it has been applied to scheduling problems [4, 5], fault diagnosis [6], function optimization [7], parameter recognition [8], etc.

However, when solving practical problems, the traditional clonal selection algorithm has problems with algorithm efficiency and convergence speed, and it easily falls into local optima [9]. The process of the clonal selection algorithm mainly includes cloning, mutation, selection, updating and other steps. At present, many scholars' improvements of clonal selection algorithms mainly focus on the optimization of key steps such as cloning, mutation and selection to improve the convergence efficiency of the algorithm. For cloning steps, Song Dan et al. [10] designed a search mechanism based on fuzzy nongenetic information, combined it with the clonal selection principle, and proposed a double clonal selection algorithm based on fuzzy nongenetic information memory, which effectively reduced the number of virtual collisions in evolutionary iteration and improved the global convergence speed and precision. Lingjie Li et al. [11] proposed a multiobjective immune algorithm cloning mechanism based on vertical distance, which was used to dynamically arrange the number of clones of each antibody to improve the population diversity. Akemi Galveza et al. [12] proposed an adaptive elite clonal selection algorithm for a multimodal and multivariate continuous nonlinear optimization problem. After the cloning steps of the algorithm were improved, cloning was no longer dependent on the affinity ratio, and all antibodies were cloned with the same probability, thus solving the optimization problem successfully. In terms of variation steps, reference [13] generates variation factors through a cloud model, improves variant antibodies, and selects clonal variant antibodies according to a reverse learning strategy to improve the global optimization ability and search efficiency of the algorithm. Yi Wang et al. [14] proposed an adaptive clonal selection algorithm with multiple differential change strategies using their historically documented pool of adaptive mutation strategies to guide the immune response process and speed up convergence through an adaptive population size adjustment method. For the mutation and selection steps, Shu Wanneng et al. [15] integrated the simulated annealing algorithm into the clonal selection algorithm, carried out mutation and selection operations according to the annealing probability, effectively maintained a balance between global search and local search, improved the search efficiency of the algorithm and accelerated the convergence rate of the algorithm.

As seen from the above literature, a large number of algorithms have been studied and used to improve cloning, mutation, selection, etc., but there is a lack

of research on the optimization of updating operation steps. The clonal selection algorithm sorts according to affinity, clones and mutates the first N antibodies with high affinity, and randomly updates the remaining d antibodies. Random updating causes problems such as blindness and uncertainty. If the position of the added antibody is close to the global optimal solution or the affinity of the added antibody is high, there is a high probability that the global optimal solution can be obtained quickly. If the added antibody is close to a local optimal solution or the antibodies gather near a local optimal solution, the probability of obtaining the global optimal solution decreases, and more iterations may be needed to obtain the global optimal solution. The position and affinity of the added antibody are random, resulting in greater randomness in the final result, and the algorithm performance is not stable. Therefore, we believe that the blind random update restricts the algorithm performance, which easily leads to slow algorithm convergence, local optimization, poor algorithm stability and other problems. To compensate for the lack of research regarding this topic, inspired by the idea of using affinity distance correlation to analyze the characteristics of optimization problems in the literature [16], we propose a new strategy to update the antibody purposefully and effectively so that the algorithm has faster convergence speed, higher optimization accuracy and better convergence stability.

This paper presents an improved clonal selection algorithm based on the directed update strategy clonal selection algorithm (DUSCSA), which focuses on improving the update operation of the clonal selection algorithm based on previous studies. In this paper, we refer to the phenomenon of antibody clustering caused by random updates of the algorithm as “crowding.” During “crowding,” the antibodies are clustered around the local optimal solution with a high probability of converging to the local optimal solution, while convergence to the global optimal solution requires more iterations. The basic idea of the algorithm is to set a minimum distance threshold to determine whether the complementary antibodies are “crowded” and to eliminate antibodies that are “crowded” and have poor affinity with the optimal antibodies in the current iteration so that the complementary antibodies are updated along a direction that is closer to the global optimal solution, increasing the chance of obtaining the global optimal solution and ensuring a stable convergence of the algorithm. The method in this paper provides a new perspective on the update phase of the clonal selection algorithm and is a useful addition to other clonal selection improvement algorithms. Our results can be transferred to improve the convergence and efficiency of other algorithms.

2 Overview of the clonal selection algorithm

2.1 Principle of the clonal selection algorithm

Organisms have a variety of immunologically active cells, and immune cells undergo mitosis to produce clones, i.e., a single cell gives rise to a group of cells. This mechanism of generating antibodies that recognize and destroy invading antigens and using asexually reproducing clones to activate differentiated and

proliferating antibodies to achieve an immune response and ultimately remove the antigen is called clonal selection [17]. Based on this mechanism, clonal selection optimization algorithms have evolved to solve function optimization problems. The correspondence between clonal selection and problem solving in biological systems is shown in Table 1.

2.2 Clonal selection algorithm flow

The flow of the traditional CSA is shown in the following steps:

Step 1: Initialize the parameter. Initialize the population size D , outstanding antibody size N , dimension dim , upper and lower bounds M and m for the array of antibody values, respectively, maximum number of iterations G_{max} , and variation probability p where $N < D$.

Step 2: Initialize the antibody population. Random generated antibody population of antibodies A of size D , x_i is the i th antibody x , $i \in [1, D]$.

$$A = \{x_1, x_2, \dots, x_i, \dots, x_D\} \quad (1)$$

Step 3: Calculate affinity. Calculate the affinity of each antibody in the antibody population by using the objective function as the target value.

Step 4: Perform segmentation. Sort the calculated antibody affinities and select the top N antibodies as the outstanding antibody population B .

$$B = \{x_1, x_2, \dots, x_i, \dots, x_N\} \quad (2)$$

Step 5: Clone. Clone each antibody at different scales, depending on affinity, to obtain the clonal population B' .

$$B' = \{x_1, \dots, x_1, x_2, \dots, x_2, x_i, \dots, x_i, \dots, x_N, \dots, x_N\} \quad (3)$$

Step 6: Mutate. Mutate the B' antibody population with probability p to obtain the mutated antibody population B'' , where x'_i is the i th mutated antibody x , $i \in [1, N]$.

Table 1 The correspondence between clonal selection algorithms and function optimization problems

Clonal selection algorithm	Function optimization problem
Antigen	Optimization problem
Antibody	A feasible solution to the optimization problem
Affinity	Quality of viable solutions
Cell activation	Immunoselection
Cell differentiation	Individual cloning
Affinity maturation	Heteromorphosis
Dynamic balance	Population update

Fig. 1 Complex function diagram

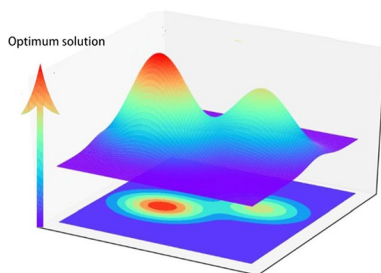
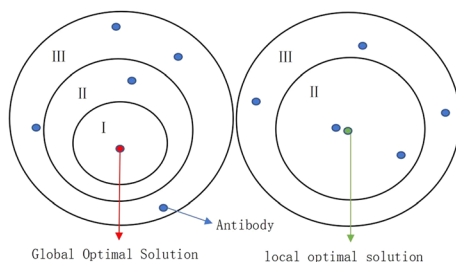


Fig. 2 Distribution of antibodies in the complex function cross section



$$B'' = \{x'_1, \dots, x'_1, x'_2, \dots, x'_2, x'_i, \dots, x'_i, \dots, x'_N, \dots, x'_N\} \quad (4)$$

Step 7: Select. For the cloned mutated antibodies, the antibody with the best affinity is selected to replace the original antibody. Finally, the new top N antibodies with good affinity are obtained as a new population of outstanding antibodies C .

$$C = \{x'_1, x'_2, \dots, x'_i, \dots, x'_N\} \quad (5)$$

Step 8: Stock update. Randomly replenish d antibodies combined with the outstanding antibody population C as the initial population for the next generation.

$$d = D - N \quad (6)$$

Step 9: Repeat step 3 if the maximum number of iterations G_{max} is not satisfied; otherwise, output the result.

2.3 Thoughts on the clonal selection algorithm

As seen from the flow of the clonal selection algorithm, after the first N good antibodies are divided in step 4, steps 5, 6 and 7 are all about cloning, mutation and selection of these N antibodies, respectively. Step 8 replaces the last d antibodies delineated in step 4 by randomly complementing them. For ease of description, as shown in Fig. 1, cross sections of a three-dimensional image of a complex function, as shown in Fig. 2, which possesses a global optimal solution and a separate local optimal solution, are taken and described by dividing them into three regions according to their affinity to the global optimal solution. The first tier comprises the antibodies with the closest affinities to the global optimal solution; the second tier

Fig. 3 Complementary incoming antibodies are clustered in the local optimal solution

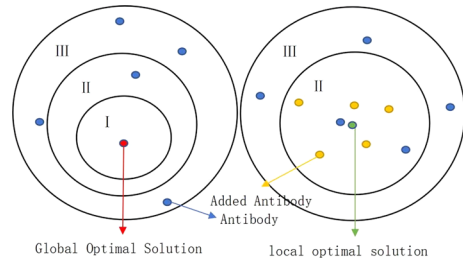
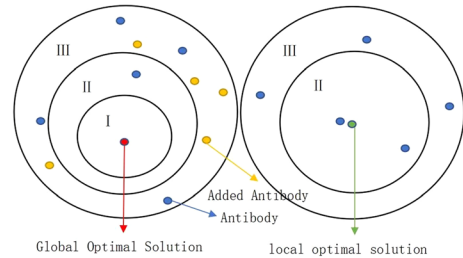


Fig. 4 Poor affinity of the supplemented antibody



comprises the antibodies with worse affinities than the first tier and better affinities than the third tier; and the third tier comprises the antibodies with the worst affinity.

With random updating, there are two possible scenarios for the distribution of the positions of the complemented antibodies, as shown in Fig. 2: first, the complemented antibodies are near the local optimal solution, and after selecting the first N antibodies and eliminating the last d antibodies, as shown in Fig. 3, the antibodies within the second layer are retained. The antibodies are all clustered near the local optimal solution, which is not conducive to obtaining the global optimal solution. Second, the antibodies that are added are near the global optimal solution, and if these antibodies are selected to participate in the next round of iterations for clonal variation, the algorithm has an increased chance of obtaining the global optimal solution. As shown in Fig. 4, the complemented antibodies have poor affinities and do not enter the second region, and most of the antibodies in the antibody pool remain clustered around the local optimal solution. Both cases result in the antibodies clustering around the local optimal solution with a high probability of converging to the local optimal solution, and convergence to the global optimal solution thus requires a higher number of iterations. Random updating is blind and uncertain; if the positions of the complementary antibodies are close to the global optimum or if the affinities of the complementary antibodies are high, there is a high probability of rapid convergence to the global optimum. Therefore, blind random updating makes the convergence of the algorithm unstable.

Based on the above analysis of the problems associated with random updating, if the randomly added antibodies are positioned close to the global optimal solution or if the added antibodies have a high affinity, the probability of obtaining the global optimal solution is higher and convergence is faster. By directing the randomly added antibodies to be updated in a direction close to the global optimum, the algorithm converges steadily and quickly, and the accuracy of the search is higher.

3 Improved clonal selection algorithm

3.1 Update operations with a targeted update policy

The directed update strategy is an improvement on the traditional random update method. The integration of the directed update strategy into the update operation of the clonal selection algorithm regulates the balance between randomness and determinism, thus avoiding the problems of low accuracy in finding the best solution and slow convergence rate due to complete randomness in the local search phase [18]. It can also avoid the blindness and uncertainty of antibodies generated by complete randomness, which may gather near a local optimal solution and produce crowding, so that the global optimum is never found.

The DUSCSA uses a threshold σ to determine whether to add a random complement of antibodies to the antibody population. The threshold calculates the farthest distance between antibodies and introduces a crowding degree factor (cdf), where $\text{cdf} \in (0, 1)$, which is used as the defining value for determining “crowding”. If the difference in position between the current antibody and the best antibody in the current iteration is greater than the threshold σ , crowding has not occurred and can be added to the antibody population; if the difference in position between the current antibody and the best antibody in the current iteration is less than the threshold σ , crowding has occurred. If the difference between the current antibody and the best antibody in this iteration is less than a threshold σ , crowding has occurred. In this case, the affinity between the two needs to be further determined, and if the current antibody has a higher affinity, this antibody can also be added to the antibody population. Otherwise, the algorithm continues to generate antibodies at random. The expression for the directed update strategy is as follows:

$$\sigma = \text{cdf} \sqrt{\sum_{i=1}^{\text{dim}} (M[i] - m[i])^2} \quad (7)$$

$$\text{div} = \sqrt{\sum_{i=1}^{\text{dim}} (x_i^g - b_i^g)^2} \quad (8)$$

where M denotes the array of upper limits of antibody values, m denotes the array of lower limits of antibody values, dim denotes the antibody dimension, div denotes the distance between the current antibody and the optimal antibody during the current iteration, x_i^g denotes the i -th variable of antibody x during the g -th iteration, b_i^g denotes the i -th variable of the optimal antibody b during the g -th iteration, $1 \leq i \leq \text{dim}$, and $g \leq G_{\max}$.

$$\begin{cases} A \Leftarrow x_i^g = r(M[i] - m[i]) + m[i] & \text{div} > \sigma \parallel \text{Aff}(x^g) > \text{Aff}(b^g) \\ r(M[i] - m[i]) + m[i] & \text{others} \end{cases} \quad (9)$$

where r denotes a uniformly distributed random number between $[0, 1]$. $\text{Aff}(x_i^g)$ denotes the affinity of the i -th variable of antibody x during the g -th iteration, and $\text{Aff}(b_i^g)$ denotes the affinity of the i -th variable of the optimal antibody b during the g -th iteration.

3.2 Improved clonal selection algorithm flow

The DUSCSA is based primarily on the clonal selection algorithm process, in which a targeted update strategy is used to determine whether an antibody is complemented or eliminated during the update phase, as shown in Fig. 5:

The implementation of the algorithm is shown in Algorithm 1:

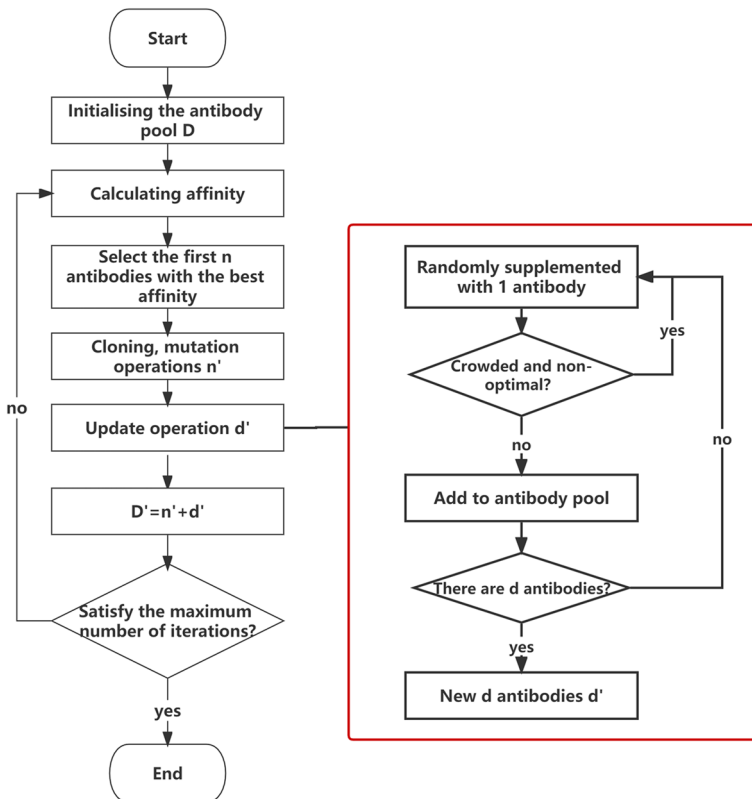


Fig. 5 DUSCSA flow

Algorithm 1 DUSCSA**Require:** $D, N, \text{dim}, M, m, G_{\max}, p, \text{cdf}$ **Ensure:** input the antibody population, D'

```

1: Initializing antibody populations:  $A = \{x_1, x_2, \dots, x_i, \dots, x_D\}$ 
2: for  $k = 0$ ;  $k < G_{\max}$ ;  $k++$  do
3:    $t = 0$ 
4:   for  $a$  in  $A$  do
5:     calculate_affinity( $a$ )
6:   end for
7:    $b^g = \text{max\_affinity}(A)$ 
8:   while  $t < d$  do
9:      $x^g = \text{Random production of antibodies}$ 
10:     $\text{div} = \sqrt{\sum_{i=0}^{\text{dim}} (x^g - b^g)^2}$ 
11:    if  $\text{div} > \sigma$  then
12:      add  $x^g$  to  $A$ ,  $t++$ 
13:    else
14:      if  $\text{affinity}(x^g) > \text{affinity}(b^g)$  then
15:        add  $x^g$  to  $A$ ,  $t++$ 
16:      else
17:        continue
18:      end if
19:    end if
20:  end while
21: end for

```

4 Improved time complexity of the clonal selection algorithm

To predict the resources required by the algorithm and the running time of the algorithm, the time complexity of the algorithm is analyzed. The time complexity can be expressed via big O notation as $O(f(n))$, where $f(n)$ is the dominant term of the complexity function. The DUSCSA contains the following main parameters: the population size D , the dimension of the antibodies dim , the clone size c , and the number of antibodies with the best affinity N . The algorithm consists of five main operational steps as follows: (1) calculation of the affinity value of the antibody; (2) cloning operation; (3) mutation operation; (4) selection operation; and (5) population directed update operation. In operation step (1), the number of executions is obtained based on the population size and is therefore D . In operation (2) and operation (3), the cloning operation on the antibody population requires cN operations. Operation (4) sorts the population to select the top N outstanding antibodies using the quicksort algorithm, which has the smallest least time complexity among sorting algorithms, and the number of operations required is $\text{clog}_2 c$. According to the definition of the update operation in this paper, operation (5) calculates the distance between the antibody and the best antibody in the current iteration process, and a minimum of d operations are required to replace the latter d antibodies. The replacement requires a comparison of the distances, and if the antibodies are judged to be crowded, the affinity between the antibody and the optimal antibody in the current iteration process also needs to be calculated. Assuming that crowding occurs for all the antibodies that are randomly added, the affinities of each need to be calculated

for further comparison, requiring a maximum number of executions of ld , where l is a constant. It is known that $D > N \geq c$ and that $D > d$. The total number of times the DUSCSA is executed during the g th iteration, S_g , should meet the following conditions:

$$S_g \leq D + cN + c \log_2 c + ld \leq 2D \quad (10)$$

Since the increased time complexity of the improved algorithm compared to the clonal selection algorithm is linear in magnitude, the DUSCSA has a maximum number of operations that is $O(D)$, which is consistent with the time complexity of the traditional clonal selection algorithm, and the update strategy does not increase the algorithm time complexity.

5 Experimental and algorithmic evaluation

In this section, 10 functions from the CEC test functions for optimization algorithms are selected to test the performance of the algorithms. The experiments are divided into two main parts: the first part compares the algorithm proposed in this paper (DUSCSA) with the classical genetic algorithm (GA) and clonal selection algorithm (CSA) to verify whether the improvements proposed in this paper are effective; the second part transposes the improvements proposed in this paper to the elite clonal selection algorithm (ECSA) proposed in the literature [12] and conducts comparison experiments with the original algorithm to verify that the proposed algorithm has an enhanced optimization effect on other clonal selection improvement algorithms. The experimental steps are as follows:

Step 1: Initialize the various parameters of the algorithm, as shown in Table 2. The genetic algorithm contains operational operators, such as crossover and variation, the clonal selection algorithm contains operational operators such as variation and cloning, and the algorithm proposed in this paper adds a crowding factor to the clonal selection algorithm. The parameters of the comparison algorithm are set according to reference [19]. The parameters of the algorithm proposed in this paper are experimentally verified, and the best performance of the algorithm is obtained when the mutation rate is set to 0.5, the initial number of clones is set to 5, and the crowding degree factor is set to 0.1.

Step 2: Set up the experimental environment, with specifics as shown in Table 3. Execute the algorithms under the same conditions and record the optimal solution

Table 2 Initialization parameters

Algorithm parameters	GA	CSA	ECSA	DUSCSA
Crossing-over rate	0.5	–	–	–
Aberration rate	0.13	0.5	2	0.5
Number of initial clones	–	5	5	5
Crowding degree factor	–	–	–	0.1

for each execution of the algorithm, running each iteration for 1000 rounds, 100 times;

Step 3: Analyze the experimental results. The first part of the experiment is analyzed in two ways: first, data analysis is performed for the experimental results. We compare the affinity for a specified number of iterations, where the smaller the affinity is, the higher the accuracy of the search, and we calculate the variance for each result, where the smaller the variance is, the better the convergence stability of the algorithm. Second, convergence curve analysis is performed, where we construct a plot with the number of iterations required to obtain a given affinity on the horizontal axis, and the affinity at a certain number of iterations on the vertical axis. The lower the number of iterations is, the faster the convergence, and the smaller the affinity is, the higher the accuracy of the search. In the second part of the experiment, the results are analyzed and compared with those the original algorithm. We compare the affinity and variance, with a smaller affinity and variance indicating that the improved step incorporating other improved algorithms improve the search accuracy and convergence stability.

5.1 Test functions

The test functions selected for this paper are shown in Table 4. Among them, $f_1 \sim f_6$ are single-peaked test functions, which have the common feature of having one global minimum, but are otherwise complex functions with multiple local minima and are suitable for testing the convergence speed and convergence stability of the detection algorithms. $f_7 \sim f_{10}$ are multi-peaked test functions, which have the common feature of having multiple global minima and multiple local minima, so they are suitable for testing the performance of the detection algorithms on high-dimensional complex functions.

5.2 Analysis of the effectiveness of the algorithm

To demonstrate the effectiveness of the improvements described in this paper for the update step of the clonal selection algorithm, comparative experiments are conducted with the classic GA and the CSA to test the algorithm's optimization accuracy, convergence speed and convergence stability by means of single-peak test functions and multi-peak test functions.

For the single-peak test functions, the optimal solution (Best), mean, standard deviation (SD) and variance of the GA, CSA and DUSCSA are recorded after 100

Table 3 Experimental environment

OS	Windows 11 professional edition
CPU	12th Gen Intel(R) Core(TM) i5-12500 H 2.50 GHz
RAM	16.0 GB
Compiler	Python 3.6

Table 4 Test functions

Test functions	Expression	Global minimum
Bukin function N. 6	$f_1(x) = 100\sqrt{ x_2 - 0.01x_1^2 + 0.01 x_1 + 10 }$	0
Drop-wave function	$f_2(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{0.5(x_1^2 + x_2^2) + 2}$	-1
Eggholder function	$f_3(x) = -(x_2 + 47) \sin\left(\sqrt{ x_2 + \frac{x_1}{2} + 47 }\right) - x_1 \sin(\sqrt{ x_1 - x_2 - 47 })$	-959.6407
Schaffer function N. 2	$f_4(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	0
Schaffer function N. 4	$f_5(x) = 0.5 + \frac{\cos(\sin(x_1^2 - x_2^2)) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	0.500094
Shubert function	$f_6(x) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i)\right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i)\right)$	-186.7309
Ackley function	$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + 20 + \exp(1)$	0
Griewank function	$f_8(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	0
Levy function	$f_9(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 (1 + 10 \sin^2(\pi w_i + 1)) + (w_d - 1)^2 (1 + \sin^2(2\pi w_d))$ where, $w_i = 1 + \frac{x_{i-1}}{4}$, for all $i = 1, \dots, d$	0
Rastrigin function	$f_{10}(x) = 10d + \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i))$	0

executions of each in the same test environment, and the experimental results are shown in Table 5.

For the multipeak test functions, we set the dimension to be $\text{dim} = 10$, $\text{dim} = 50$ and $\text{dim} = 100$, and record the optimal solution (Best), mean, standard deviation (SD) and variance. The experimental results for each dimension are shown in Tables 6, 7 and 8.

The analysis of the data in Table 5 shows that of the three algorithms, the CSA has the best search accuracy on f_1 , and the DUSCSA has the best search accuracy and convergence stability on f_2 , f_3 , f_4 , f_5 , and f_6 . The experimental variance of DUSCSA on f_3 , f_4 and f_5 is 0, and the convergence is particularly stable. From the data in Table 6, when the dimension is 10, the GA has the best optimization accuracy on f_9 , the CSA has the best optimization accuracy on f_{10} , and the DUSCSA has the best convergence stability on f_7 . In the experimental results of f_8 , the search accuracy of the DUSCSA is better than those of the GA and the CSA. From the data in Table 7, when the dimension is 50, the GA has the best search accuracy on f_8 , the CSA has the best search accuracy on f_9 and f_{10} , and the DUSCSA has the best accuracy and still maintains good convergence stability on f_7 . From the data in Table 8, the optimization accuracy of the GA is the on f_9 , the optimization accuracy of the CSA is the best on f_{10} , and the optimization accuracy of the DUSCSA is the best on f_7 and f_8 . Compared with the GA and the CSA, the DUSCSA has better convergence stability when the dimension is 100. Convergence graphs provide a more intuitive

Table 5 Single-peak test function experimental results

Function	Algorithm	Best	Mean	SD	Variance
f_1	GA	1.8930E-01	4.1033E-01	3.1400E-01	5.6400E-02
	CSA	1.6300E-02	1.1029E-01	5.0314E-01	9.1270E-02
	DUSCSA	2.0000E-02	7.4000E-02	3.6080E-02	1.3000E-03
f_2	GA	3.6700E-13	2.6222E-10	5.0223E-10	4.9211E-18
	CSA	2.3700E-16	1.6205E-14	3.5404E-14	1.2534E-27
	DUSCSA	3.1300E-17	8.4879E-15	3.2541E-14	1.0589E-27
f_3	GA	-9.4500E+02	-9.3500E+02	1.4200E-02	2.3200E-03
	CSA	-9.5960E+02	-9.5915E+02	3.4370E+00	1.1813E+01
	DUSCSA	-9.5964E+02	-9.5964E+02	00000E+00	00000E+00
f_4	GA	1.8900E-04	2.0100E-04	6.4100E-05	5.3200E-06
	CSA	2.0100E-04	2.7700E-03	3.9800E-03	3.1168E-07
	DUSCSA	00000E+00	00000E+00	00000E+00	00000E+00
f_5	GA	5.0000E-01	3.1222E-01	3.4900E-06	5.0872E-16
	CSA	5.0000E-01	3.8920E-01	6.3600E-09	4.0449E-17
	DUSCSA	5.0000E-01	5.0000E-01	00000E+00	00000E+00
f_6	GA	-1.6700E+02	-1.6500E+02	2.9000E-03	3.2129E-04
	CSA	-1.8700E+02	-1.8667E+02	1.1290E-03	1.3348E-05
	DUSCSA	-1.8700E+02	-1.8669E+02	2.9100E-04	3.3482E-06

Table 6 Multimodal test function experimental results (dim = 10)

Function	Algorithm	Best	dim = 10		
			Mean	SD	Variance
f_7	GA	2.3931E-01	1.2838E+01	6.2993E+00	2.5098E+00
	CSA	1.1482E-01	2.0213E+01	1.0049E+01	3.1700E+00
	DUSCSA	4.4409E-16	6.8000E+00	3.4000E+00	1.8439E+00
f_8	GA	2.3931E-01	6.3800E-01	1.9935E-01	4.4648E-01
	CSA	1.7754E-01	3.8669E-01	1.0458E-01	3.2338E-01
	DUSCSA	1.2026E-01	3.5272E-01	1.1623E-01	3.4093E-01
f_9	GA	9.7106E-01	1.0947E+00	6.1820E-02	2.2764E-01
	CSA	9.8106E-01	1.0095E+00	1.4220E-02	1.1925E-01
	DUSCSA	9.9106E-01	9.9980E-01	4.3700E-03	1.0072E-01
f_{10}	GA	9.2171E+00	9.2512E+00	1.7060E-02	1.3061E-01
	CSA	7.2359E+00	7.3241E+00	4.4100E-02	2.1000E-01
	DUSCSA	7.3700E+00	7.3733E+00	1.6405E-03	4.0503E-02

picture of each algorithm's performance. Figures 6, 7, 8, 9, 10 and 11 show the convergence curves of the DUSCSA compared to those of the CSA for the six single-peak test functions.

Table 7 Multimodal test function experimental results (dim = 50)

Function	Algorithm	Best	dim = 50		
			Mean	SD	Variance
f_7	GA	2.1831E-01	1.1838E+01	5.8098E+00	2.5098E+00
	CSA	1.0005E-01	1.9442E+01	9.6710E+00	3.1700E+00
	DUSCSA	4.8941E-16	6.4000E+00	3.2000E+00	1.8439E+00
f_8	GA	5.1000E-01	1.8380E+00	6.6400E-01	4.4648E-01
	CSA	8.1000E-01	1.8104E+00	5.0020E-01	3.2338E-01
	DUSCSA	8.5000E-01	1.2174E+00	1.8370E-01	3.4093E-01
f_9	GA	9.9514E-01	1.3022E+00	1.5353E-01	2.2764E-01
	CSA	9.5139E-01	1.0330E+00	4.0805E-02	1.1925E-01
	DUSCSA	9.7951E-01	9.8146E-01	9.7500E-04	1.0072E-01
f_{10}	GA	9.5263E+01	1.2760E+02	1.6169E+01	1.3061E-01
	CSA	7.5263E+01	1.1705E+02	2.0894E+01	2.1000E-01
	DUSCSA	8.4033E+01	1.1206E+02	1.4014E+01	4.0503E-02

Table 8 Multimodal test function experimental results (dim = 100)

Function	Algorithm	Best	dim = 100		
			Mean	SD	Variance
f_7	GA	1.8393E-01	1.0558E+01	5.1870E+00	2.5098E+00
	CSA	1.1482E-01	1.8332E+01	9.1086E+00	3.1700E+00
	DUSCSA	4.4409E-16	5.2355E+00	2.6178E+00	1.8439E+00
f_8	GA	5.2211E+00	5.9281E+00	3.5350E-01	4.4648E-01
	CSA	5.1600E+00	6.2156E+00	5.2780E-01	3.2338E-01
	DUSCSA	4.6200E+00	5.1456E+00	2.6280E-01	3.4093E-01
f_9	GA	9.5139E-01	1.2899E+00	1.6926E-01	2.2764E-01
	CSA	9.7951E-01	1.2899E+00	1.5520E-01	1.1925E-01
	DUSCSA	9.9514E-01	9.9915E-01	2.0030E-03	1.0072E-01
f_{10}	GA	4.9820E+02	1.3621E+03	4.3195E+02	1.3061E-01
	CSA	4.5820E+02	1.2831E+03	4.1245E+02	2.1000E-01
	DUSCSA	5.3714E+02	1.2629E+03	3.6288E+02	4.0503E-02

For the single-peak test functions, the DUSCSA has more stable and reliable convergence and better accuracy than the GA and CSA. For the multi-peak test functions, the optimization accuracy of all three algorithms decreases as the dimensionality increases, but the DUSCSA's optimization accuracy is less affected by the dimensionality and maintains good convergence stability. The experimental results show that the update strategy proposed in this paper is effective, and the algorithm has good robustness and ensures stable convergence. The comparative graphs of the convergence curves of the six tested functions clearly

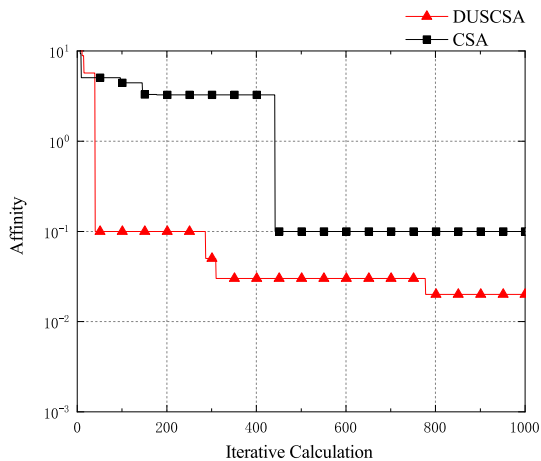
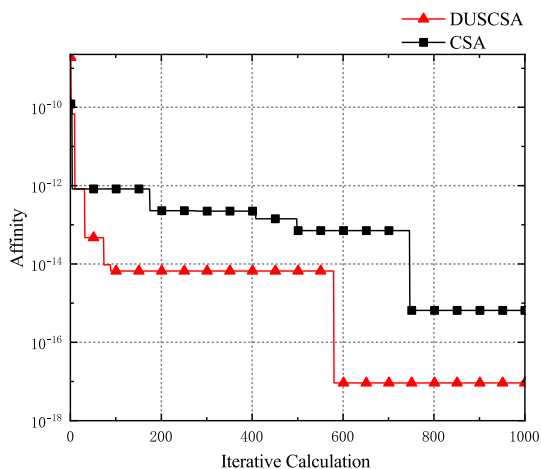
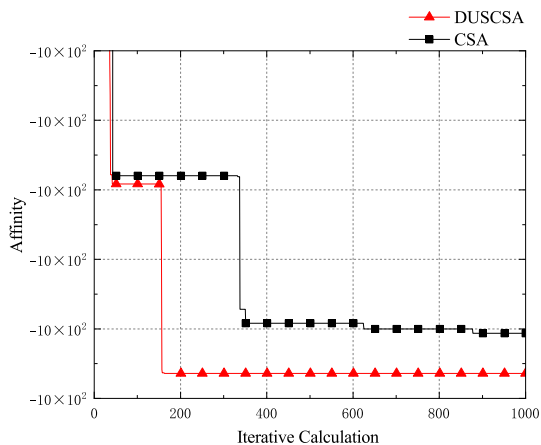
Fig. 6 Convergence curves of f_1 **Fig. 7** Convergence curves of f_2 **Fig. 8** Convergence curves of f_3 

Fig. 9 Convergence curves of f_4

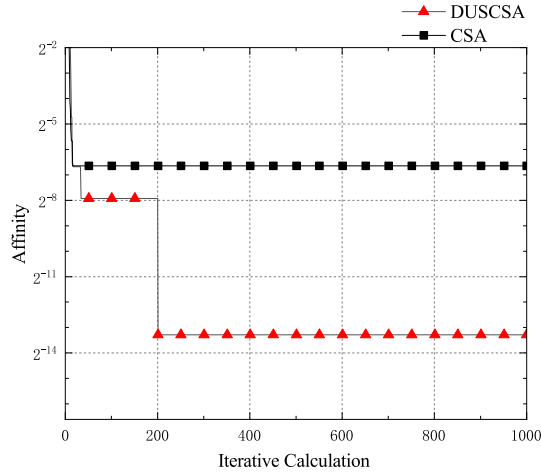


Fig. 10 Convergence curves of f_5

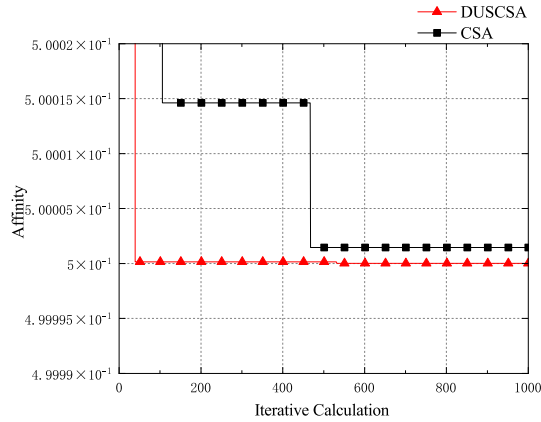
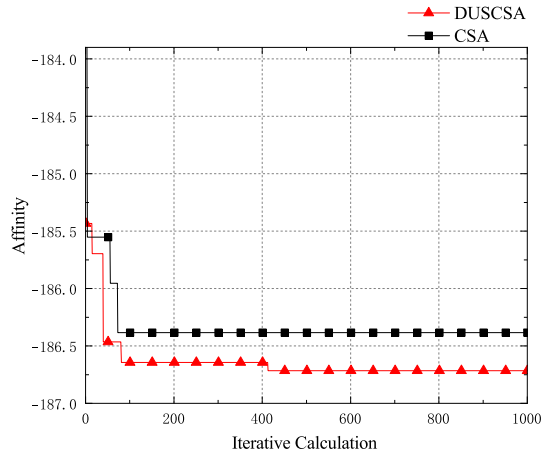


Fig. 11 Convergence curves of f_6



show that the DUSCSA is more accurate than the CSA under the same number of iterations; the DUSCSA has fewer iterations than the CSA under the same affinity and therefore converges faster. The comparison of the results reveals that the overall performance of the DUSCSA is 1% higher than that of the CSA and 2.2% higher than that of the GA. In summary, the update strategy proposed in this paper is an effective way to improve the algorithm's search accuracy and convergence stability.

5.3 Analysis of the portability of the algorithm

To demonstrate that the improvements to the update step can be used in other clonal selection improvement algorithms, the improvements in the update phase were fused into the ECSA. We compared this fused algorithm with the ordinary ECSA and recorded the optimal solution (Best), mean, standard deviation (SD) and variance after 100 executions of each in the same test environment. The experimental results are shown in Table 9.

The experimental results for f_1 show that the ECSA has the higher accuracy, but among the remaining test functions, the ECSA + DUSCSA has better accuracy and convergence stability than the original algorithm. Among these functions, the experimental variance of f_3 , f_4 and f_5 is 0, and the convergence is particularly stable. The overall performance of the ECSA + DUSCSA is better than that of the ECSA by 0.97%, indicating that the improvements proposed in this paper can be transferred to other clone selection improvement algorithms and further enhance their search accuracy and convergence stability.

Table 9 Comparison of experimental results with the improved algorithm

Function	Algorithm	Best	Mean	SD	Variance
f_1	ECSA	1.6600E-02	1.1429E-01	5.2314E-01	9.9270E-02
	ECSA + DUSCSA	2.0300E-02	6.4000E-02	4.6080E-02	2.3000E-03
f_2	ECSA	2.3100E-16	1.2205E-14	3.3404E-14	1.4534E-27
	ECSA + DUSCSA	2.1450E-17	6.4878E-15	2.6541E-14	1.8839E-27
f_3	ECSA	-9.5560E+02	-9.5415E+02	3.4330E+00	1.1823E+01
	ECSA + DUSCSA	-9.5966E+02	-9.5966E+02	00000E+00	00000E+00
f_4	ECSA	2.0600E-04	2.7500E-03	3.9840E-03	3.1368E-07
	ECSA + DUSCSA	00000E+00	00000E+00	00000E+00	00000E+00
f_5	ECSA	5.0000E-01	3.8929E-01	6.3608E-09	4.0447E-17
	ECSA + DUSCSA	5.0000E-01	5.0000E-01	00000E+00	00000E+00
f_6	ECSA	-1.8699E+02	-1.8661E+02	1.1292E-03	1.3348E-05
	ECSA + DUSCSA	-1.8700E+02	-1.8669E+02	2.8100E-04	3.4782E-06

6 Conclusion

Many clonal selection algorithms have been studied, and improvements have been made to the cloning, mutation, and selection steps, but there is a lack of research on optimizing the update operation. In this paper, a clonal selection algorithm based on a directional update strategy is proposed to address the blindness and uncertainty of the random update of the clonal selection algorithm, which is responsible for a high number of iterations, low optimization accuracy, and poor convergence stability. If the distance is less than the threshold and the affinity is worse than that of the optimal antibody in the current iteration, then the antibody is readded to increase the probability of finding the global optimal solution and improve the convergence speed and stability of the algorithm. If the randomly added antibodies are located close to the global optimum or have high affinity, the probability of obtaining the global optimum is higher, and convergence is faster. By guiding the randomly added antibodies toward the global optimum, the algorithm converges steadily and quickly, and the accuracy of the search is higher. The evolution of the antibodies toward the global optimum is more in line with the nature of biological evolution.

The experiments demonstrate that the improvement proposed in this paper is an effective updating strategy that improves the accuracy, enhances the convergence stability and increases the convergence speed of the clonal selection algorithm. The strategy actually uses the Euclidean distance formula to calculate the similarity for determination, and more methods for calculating the similarity can be chosen for different comparison experiments to select an optimal way for improvement. The improvement proposed in this paper not only addresses the lack of research on the update phase of the clonal selection algorithm but can also be applied to other heuristic algorithms that need to perform population updates, further enhancing their optimization accuracy and convergence stability.

Author contributions CY proposed the main structure of the algorithm in this paper, evaluated its feasibility, and proposed the flow of the algorithm in this paper. ZH performed the coding of algorithms, the collation of test functions, and the experiments of related algorithms. BJ participated in the feasibility demonstration of the algorithm and helped perform optimization of the algorithm. MZ helped analyze the algorithm time complexity and provided experimental methods. AL made great contributions in the experimental stage, including debugging after code writing and optimizing the code so that the algorithm could obtain results faster. JH helped organize the experimental datasets and record the experimental results during the experimental stage and participated in the analysis stage of the final experimental results.

Funding This work is partially supported by the National Natural Science Foundation of China (61977021) and the Hubei Provincial Key R&D Program Project (2021BAA184).

Availability of data and materials This article uses 10 optimization functions from the CEC benchmark functions.

Code availability Relevant codes can be obtained by contacting cercischinensis@foxmail.com.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

Ethical approval All experimental data in this paper are real and valid, and there is no case of multiple submissions for one manuscript in this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Yan X, Li P, Tang K, Gao L, Wang L (2020) Clonal selection based intelligent parameter inversion algorithm for prestack seismic data. *Inf Sci* 517:86–99. <https://doi.org/10.1016/j.ins.2019.12.083>
2. Wang D, Liang Y, Dong H, Tan C, Xiao Z, Liu S (2022) Innate immune memory and its application to artificial immune systems. *J Supercomput* 78:1–22. <https://doi.org/10.1007/s11227-021-04295-1>
3. Chawla VK, Chanda AK, Angra S (2019) A clonal selection algorithm for minimizing distance travel and back tracking of automatic guided vehicles in flexible manufacturing system. *J Inst Eng (India) Ser C* 100:401–410
4. Yanming QUANHY (2021) Research on clonal selection algorithm for multi-robot task allocation and scheduling. *J S China Univ Technol (Nat Sci Edn)* 49(5):102–110
5. Cheikhrouhou O, Khoufi I (2021) A comprehensive survey on the multiple traveling salesman problem: applications, approaches and taxonomy. *Comput Sci Rev* 40:100369. <https://doi.org/10.1016/j.cosrev.2021.100369>
6. Yu-e G (2017) Research on fault recovery of ship distribution network based on cloning algorithm. *Ship Sci Technol* 39(16):163–165
7. Zhang FB, Fan XL (2018) Research on multi-objective optimal clone selection algorithm for immune intrusion detection. *Comput Eng Sci* 40(02):261–267
8. Chen Q, Fu CQP (2022) Parameter identification of permanent magnet synchronous motor based on clone-selective differential evolutionary algorithm. *Sens Microsyst* 41(01): 135–137141. [https://doi.org/10.1387/J.1000-9787\(2022\)01-0135-03](https://doi.org/10.1387/J.1000-9787(2022)01-0135-03)
9. Luo W, Lin X (2018) Recent advances in clonal selection algorithms and applications. *Comput Intell* 1:12–2. <https://doi.org/10.1109/SSCI.2017.8285340>
10. Song D, Fan X, Wen Z, Huang D, Qu X (2017) Double clonal selection algorithm based on fuzzy non-genetic information memory. *J Electron Inf* 39(02):255–262. <https://doi.org/10.11999/JEIT160359>
11. Li L, Lin Q, Li K, Ming Z (2021) Vertical distance-based clonal selection mechanism for the multiobjective immune algorithm. *Swarm Evolut Comput* 63:100886. <https://doi.org/10.1016/j.swevo.2021.100886>
12. Gálvez A, Iglesias A, Avila A, Otero C, Arias R, Manchado C (2015) Elitist clonal selection algorithm for optimal choice of free knots in b-spline data fitting. *Appl Soft Comput* 26:90–106. <https://doi.org/10.1016/j.asoc.2014.09.030>
13. Wanf LL, Shen Y, Xu YS et al (2021) A clone selection algorithm incorporating cloud models and backward learning. *Comput Eng Appl* 57(17):68–74
14. Wang Y, Li T, Liu X, Yao J (2022) An adaptive clonal selection algorithm with multiple differential evolution strategies. *Inf Sci* 604:142–169. <https://doi.org/10.1016/j.ins.2022.04.043>
15. Shu WN, Ding LX (2016) Optimization and quality factors of clone selection algorithms. *J Softw* 27(11):2763–2776. <https://doi.org/10.13328/j.cnki.jos.004911>

16. Tan Z, Tang Y, Li K, Huang H, Luo S (2022) Differential evolution with hybrid parameters and mutation strategies based on reinforcement learning. *Swarm Evol Comput* 75:101194. <https://doi.org/10.1016/j.swevo.2022.101194>
17. Zhang R-C, Pan C-Y, Wu X, Yang K (2021) Adaptive sampling immune optimization algorithm for nonlinear multi-objective probabilistic optimization problems. *Acta Electronica Sinica* 49(04):647–660. <https://doi.org/10.12263/DZXB.20200171>
18. Liu J-S, Ma Y-X, Li Y (2021) Improved butterfly algorithm for multi-dimensional complex function optimization problem. *Acta Electronica Sinica* 49(06):1068–1076
19. Yang C, Chen B-Q, Jia L, Wen H-Y (2020) Improved clonal selection algorithm based on biological forgetting mechanism. *CompLex* 2020:2807056–1280705610

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.