



# A Chaotic Antlion Optimization Algorithm for Text Feature Selection

Hongwei Chen<sup>1</sup> · Xun Zhou<sup>1</sup> · Dewei Shi<sup>1</sup>

Received: 28 December 2021 / Accepted: 1 June 2022  
© The Author(s) 2022

## Abstract

Text classification is one of the important technologies in the field of text data mining. Feature selection, as a key step in processing text classification tasks, is used to process high-dimensional feature sets, which directly affects the final classification performance. At present, the most widely used text feature selection methods in academia are to calculate the importance of each feature for classification through an evaluation function, and then select the most important feature subsets that meet the quantitative requirements in turn. However, ignoring the correlation between the features and the effect of their mutual combination in this way may not guarantee the best classification effect. Therefore, this paper proposes a chaotic antlion feature selection algorithm (CAFSA) to solve this problem. The main contributions include: (1) Propose a chaotic antlion algorithm (CAA) based on quasi-opposition learning mechanism and chaos strategy, and compare it with the other four algorithms on 11 benchmark functions. The algorithm has achieved a higher convergence speed and the highest optimization accuracy. (2) Study the performance of CAFSA using CAA for feature selection when using different learning models, including decision tree, Naive Bayes, and SVM classifier. (3) The performance of CAFSA is compared with that of eight other feature selection methods on three Chinese datasets. The experimental results show that using CAFSA can reduce the number of features and improve the classification accuracy of the classifier, which has a better classification effect than other feature selection methods.

**Keywords** Antlion optimization · Chaotic map · Feature selection · Quasi-opposition learning strategy · Text classification

## Abbreviations

CAA	Chaotic antlion algorithm
CAFSA	Chaotic antlion feature selection algorithm
GWO	Gray wolf optimizer
SVM	Support vector machines
ALO	Ant lion optimizer
QOBL	Quasi-opposition based learning
PWLCM	Piecewise linear chaotic map
OBL	Opposition based learning
DA	Dragonfly algorithm
SCA	Sine cosine algorithm
IALOT	An improved multiobjective antlion optimization algorithm

TFIDF	Term frequency–inverse document frequency
CALOP	Piecewise linear chaotic antlion optimization
SOA	Seagull optimization algorithm
SSA	Sparrow search algorithm
HHO	Harry Hawk optimizer
NB	Naive Bayes
DT	Decision tree

## 1 Introduction

Due to the continuous advancement of Internet technology and the increasing scale of scientific computing, the amount of various electronic text data has increased sharply, and the text data is neither completely unstructured nor completely structured. The dimensionality of a text vector can generally be as high as tens of thousands of dimensions. Therefore, in the data mining and data retrieval process, the feasibility is often reduced due to excessive calculation or high cost. As a classic task type in the field of text data mining, text classification has attracted the attention of numerous researchers. It has been frequently applied in real-life scenarios such

✉ Xun Zhou  
void37@qq.com  
Hongwei Chen  
chw2001@sina.com  
Dewei Shi  
sdw1901@163.com

<sup>1</sup> School of Computer Science, Hubei University of Technology, Nan Li Road, Wuhan 430068, China

as e-commerce reviews, spam filtering, sentiment analysis, intent recognition, and personalized recommendation [1]. In the process of text classification, a document usually contains thousands of unique words, and each word is regarded as a feature. But many features may contain a small amount of information, be noisy, or have nothing to do with category labels. This may reduce the performance and accuracy of the classification models at a later stage and increase the computational complexity [2]. Therefore, one of the main problems faced by text classification is to deal with high-dimensional feature sets.

Feature selection is a method of extracting important or superior features from a high-dimensional feature set. After data preprocessing, feature selection is used to control the dimensionality of the feature space to eliminate redundant features, thereby improving the efficiency and accuracy of the classifier while reducing the complexity of calculation [3]. Feature selection is a combinatorial problem, and bionic algorithm is a classic means to solve optimization problems and combinatorial problems. Moreover, before performing feature-related processing in various fields, it is generally necessary to binarize features, such as text classification [4], image recognition [5], and cancer prediction [6]. In 2015, Australian Professor Mirjalili proposed the ant lion optimizer (ALO) [7]. Because the algorithm has the advantages of fewer adjustment parameters and simple implementation, some scholars have successfully applied the antlion algorithm to the field of feature selection. However, it is rarely used to solve the text feature selection problem. Therefore, in this study, we propose a chaotic antlion optimization algorithm for text feature selection.

The main contributions of this paper include: (1) First, propose a chaotic antlion algorithm (CAA) based on a quasi-opposite learning mechanism and a chaos strategy, and compare the optimization performance with four algorithms on eleven benchmark functions. (2) Study the performance of CAFSA using CAA for feature selection when using different learning models, including decision tree, Naive Bayes, and SVM classifier. (3) The performance of CAFSA is compared with that of eight other feature selection methods on three Chinese datasets. Experimental results show that CAFSA is better than the other eight algorithms.

The other chapters of this article are arranged as follows: Sect. 2 is a summary of related work. Section 3 briefly introduces the process of the native antlion optimization algorithm, and Sect. 4 introduces the improvement strategies used in CAA, as well as the algorithm process and performance test experiments. Section 5 introduces the process of CAFSA using CAA in feature selection for Chinese text classification. Section 6 gives the comparison of CAFSA algorithm and other feature selection algorithms on three Chinese data sets when using different classifiers. Section 7 is a summary of the work of this article and future prospects.

## 2 Related Work

Feature selection techniques are usually divided into three categories: embedded methods, filter methods and wrapper methods [8]. Among them, the embedded method is a combination of the latter two methods. Although the filter method based on variable correlation has good generalization ability and low computational complexity, it may not be able to extract the best subset [9], similar to MICIMR [10], MIM [11], MIFS [12] MRMR [13] and TCIC\_FS [14] and so on. Compared with filter methods, wrapper algorithms are more accurate because they consider the relationship between features [15]. Wrapper feature selection methods using a biomimetic algorithm show great possibilities for efficiently handling high-dimensional feature vectors [16].

In addition to the classic particle swarm optimization [17–19] and genetic algorithm [20], some other novel bionic algorithms have been successfully applied to text feature selection. For example, the cat swarm optimization algorithm [21], artificial fish swarm algorithm [22], the Jaya optimization algorithm [23], the firefly algorithm [24], the grey wolf optimization algorithm [25] and the ant colony algorithm [26]. To solve the problem of Arabic text classification, Chantar et al. proposed an enhanced binary gray wolf optimizer (GWO) as a feature selection method [27]. The method is combined with the SVM classifier and experiments are performed on three Arabic data sets. The results show that the scheme has superior performance on the problem of Arabic document classification. Kyaw et al. uses natural-inspired heuristic methods (whale and gray wolf optimization algorithm) and chi-square algorithm for feature selection, which improves the classification accuracy of the naive Bayes classifier on the 20 News Group data set [28]. To sum up, there have been a lot of research on text feature selection algorithms for English and Arabic, but they are relatively lacking in the Chinese field, especially bionic feature selection algorithms.

At present, some scholars have successfully applied the antlion algorithm to the field of feature selection. Mingwei Wang et al. used wavelet support vector machine (WSVM) to enhance the stability of the classification results, and at the same time used Lévy flight to help the swarm intelligence algorithm jump out of the local optimum [29]. Li Mengmeng et al. proposed a stable ant-antlion optimiser (SALO), which combines ant colony optimisation and ALO, and compared it with several relatively stable filtering methods [30]. Zawbaa et al. proposed a hybrid bio-inspired heuristic approach, combining ALO with grey wolf optimization (GWO) and comparing it with GA and PSO [31]. In conclusion, these proposed improved antlion algorithms are either combined with other bionic algorithms or improved native ALO, and then compared with genetic algorithm and particle swarm

algorithm and some classical filter feature selection algorithms on different datasets, which shows the superiority of the ALO algorithm in feature selection. However, the original antlion optimization algorithm has disadvantages such as slow convergence speed, low accuracy of solving multimodal functions, and tend to get stuck in local optimal solutions. Therefore, scholars have made various refinements in response to these shortcomings. Yao et al. [32] used the Levy flight mode and combined the 1/5 principle in the closed-loop system of the control system theory to dynamically adjust the shrinkage rate of the trap, making the search range of the algorithm more comprehensive and the overall optimization efficiency higher. Saha et al. [33] used the quasi-opposite learning mechanism to initialize the population and the CLS strategy to improve the update formula for the ant position. Kilic et al. [34] replaced the roulette selection method with the tournament strategy and reduced the scale of the random walk of the ant population, which contributes to improving the accuracy and speed of the algorithm. Guo et al. proposed an improved antlion optimizer (OB-C-ALO) [35] and used it to solve the data clustering problem. The optimizer uses a random walk based on the Cauchy distribution rather than a uniform distribution to escape from local optimal solutions the local optimal value, and then combines the learning model based on the opposition with the acceleration coefficient, which overcomes the shortcomings of the slow convergence of the original ALO. Zawbaa et al. [36] uses different chaotic maps to change the reduction ratio of the antlion trap boundary, which improves the algorithm's optimization ability.

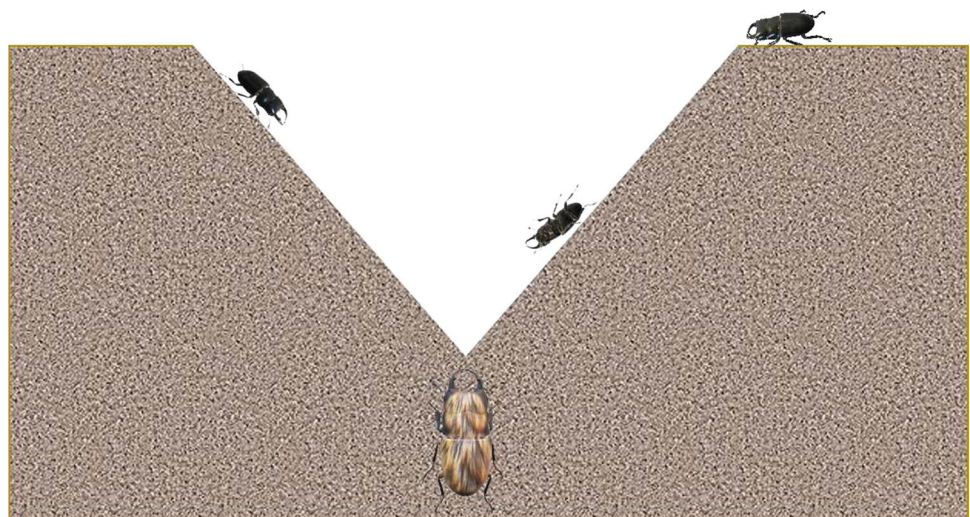
Although these improvements have achieved good results in their respective fields, there is still a lot of room for improvement in the performance of ALO when used in text feature selection. The traditional ALO generates the initial population randomly, which cannot guarantee the quality of

the initial population and may lead to some groups learning from poor individuals, reducing the convergence speed of the algorithm. Besides, the model and scale of the random walk of the ant colony are two main reasons about why the algorithm is easy to become stuck in the local optimum solution and runs slowly. To this end, this paper proposes an improved ALO feature selection algorithm based on quasi-dissenting opinion learning mechanism and chaotic mapping, and studies the effect of this method compared with other eight feature selection methods when combined with different classifiers.

### 3 Ant Lion Optimization Algorithm

The ant lion optimizer finds the optimal solution by emulating the process of antlions building traps and preying on ants in nature. Antlions usually rotate around the sand and drill down to make a funnel-shaped trap. Then they hide under the sand at the bottom of the funnel and use their big jaws to bounce the sand out to make the funnel smooth and steep. When ants or other insects crawl with the trap, they will slide down as the sand loosens. At the same time, to prevent the prey from escaping, the antlions continue to bounce the sand outwards, causing the prey to be pushed into the center by the quicksand. The antlion algorithm has two populations, one for ants and one for antlions. Ants walk randomly, ants fall into traps, ants slide towards antlions, antlions catch ants and antlions rebuild the trap are the five fundamental modules of the hunting process. Figure 1 depicts the natural hunting activity of antlions.

**Fig. 1** Hunting behavior of antlions



### 3.1 Random Walk of Ants

$$\chi(t) = [0, \text{Sum}(2\gamma(t_1) - 1), \text{Sum}(2\gamma(t_2) - 1), \dots, \text{Sum}(2\gamma(t_n) - 1)], \quad (1)$$

where Sum calculates the cumulative sum,  $t$  represents the current number of iterations, and  $n$  represents the maximum number of iterations. Among them, rand is a random number uniformly distributed between 0 and 1.

$$\gamma(t) = \begin{cases} 1, & \text{if rand} > 0.5 \\ 0, & \text{if rand} \leq 0.5. \end{cases} \quad (2)$$

To guarantee that ants move randomly in the search space and do not violate any boundaries, the position of ants needs to be regulated:

$$\chi_i^t = \frac{(\chi_i^t - a_i) \times (ub_i^t - lb_i^t)}{b_i - a_i} + lb_i^t \quad (3)$$

$lb_i^t$  and  $ub_i^t$  are the minimum and maximum values of the  $i$ -th variable of a certain ant in the  $t$ -th generation.  $a_i$  and  $b_i$  are the minimum and maximum values of the  $i$ th variable of all ants.

### 3.2 Fall into the Trap

By simulating the process of ants walking aimlessly around the antlion and becoming trapped, the calculation formula is as follows:

$$\begin{aligned} lb_i^t &= \text{Antlion}_j^t + lb^t \\ ub_i^t &= \text{Antlion}_j^t + ub^t \end{aligned} \quad (4)$$

### 3.3 Ants Sliding Towards Antlions

By adaptively lowering the ant's random walk range, the process of simulating the ants sliding into the antlions is as follows:

$$\begin{aligned} lb^t &= \frac{lb^t}{I} \\ ub^t &= \frac{ub^t}{I}, \end{aligned} \quad (5)$$

where  $t$  is the current number of iterations,  $T$  is the maximum number of iterations, and dynamically adjust  $I = 1 + 10^{\nu \times \frac{t}{T}}$  according to the increase in the number of iterations.  $\nu$  depends on the current number of iterations.

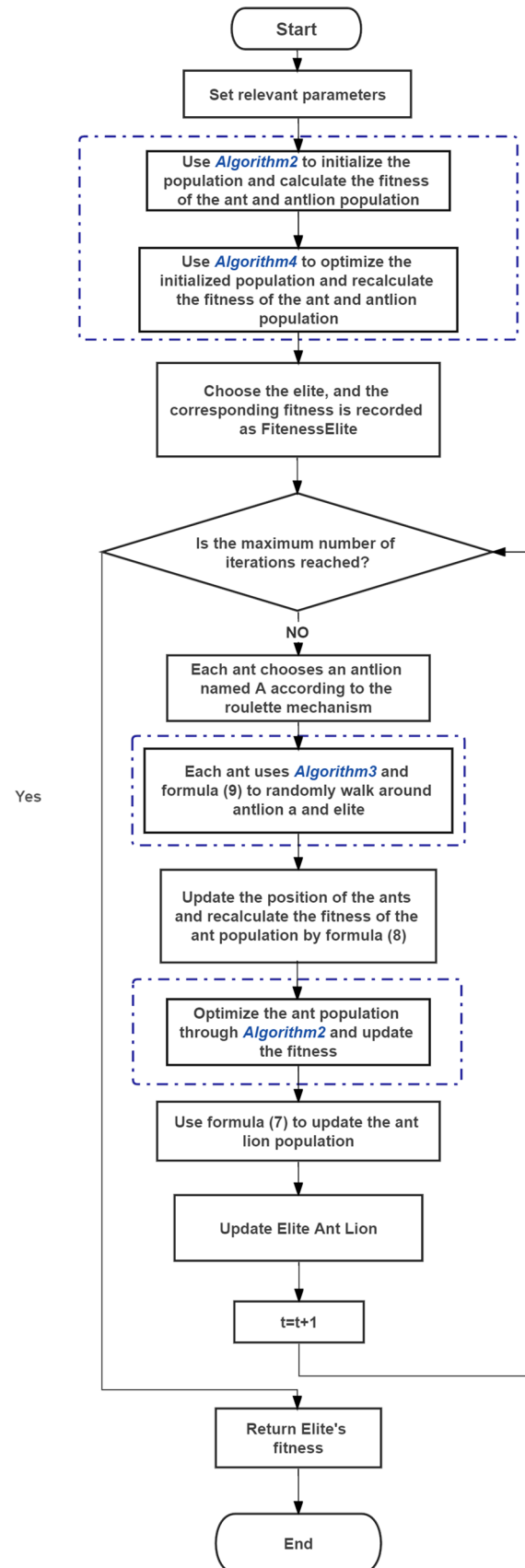


Fig. 2 Flow chart of CAA algorithm



$$t > \begin{cases} 0.1T & v = 2 \\ 0.5T & v = 3 \\ 0.75T & v = 4 \\ 0.9T & v = 5 \\ 0.95T & v = 6 \end{cases} \quad (6)$$

### 3.4 Ants are Caught

$$\text{Antlion}_j^t = \text{Ant}_i^t, \text{ if } f(\text{Ant}_i^t) < f(\text{Antlion}_j^t), \quad (7)$$

where  $\text{Ant}_i^t$  is the position of the  $i$ -th ant in the  $t$ -th iteration, and similarly,  $\text{Antlion}_j^t$  is the position of the  $j$ -th antlion at the  $t$ -th iteration. Update the position of the ants with the following formula:

$$\text{Ant}_i^t = \frac{R_i^t + R_E^t}{2} \quad (8)$$

$R_i^t$  is the position obtained by the ant moving randomly around the antlion selected by the roulette wheel selection mechanism and  $R_E$  is the position obtained by the ants moving randomly around the elite antlion at the  $t$ -th iteration.

### 3.5 Antlion Rebuild Trap

Calculate the updated objective function value of the ant, and compare it with the elite, and choose a better value as

the global optimal solution. If the fitness value of the ant is less than the antlions, it is considered that the antlions have successfully captured the ant and the position of the antlions is updated.

## 4 The Chaotic Antlion Algorithm (CAA)

Figure 2 is a flowchart of the chaotic antlion algorithm(CAA). CAA first uses the PWLCM chaotic map to initialize the positions of ants and antlions, and then uses a quasi-opposite learning strategy to optimize the positions of the two, increasing the diversity of the population. Then, the random walk behavior of the ants uses the chaotic strategy to increase the search range, so that the algorithm can jump out of the local optimal solution. At the same time, the walking scale of ants is reduced according to the principle of 1/5, which speeds up the chaotic walking speed of ants. Finally, after the ants are successfully captured, the quasi-opposite learning strategy is used again to optimize the position of the antlion, which improves the accuracy of the algorithm optimization. The pseudocode for CAA is described in detail in Algorithm 1. The following subsections detail the different components of the algorithm.

### 4.1 Reduce the Size of Random Walk

---

**Algorithm 1** The pseudo-code of the chaotic antlion algorithm

---

**Input:** Setting related parameters includes group size  $N$ , individual dimensions  $d$ , boundaries  $[lb, ub]$ , chaos parameters  $\varphi, t = 0$ ,

**Output:** The best fitness  $Fitness_{Elite}$  of the Elite antlion

```

1: Use Algorithm.2 to initialize the location of ants and antlions and then use
   Algorithm.4 to optimize, and calculate the fitness of the population
2: The antlion with the best fitness is selected as Elite, and the corresponding fitness
   is recorded as  $Fitness_{Elite}$ 
3: while  $t \leq MaxIter$  do
4:   for each ant do
5:      $Antlion_A$  obtained by roulette mechanism;
6:     for every dimension do
7:       Use Eq.(6) to adaptively adjust the boundary reduction ratio  $I$ 
8:       Use Eq.(4) and Eq.(5) to update the boundary
9:       Using Algorithm3 to generate the random walk  $RW$ 
10:      Using the Eq.(9) to generate the  $RA$  and  $RE$ .
11:     end for
12:   end for
13:   Update the position of the ants through Eq.(8)
14:   Optimize the ant population through Algorithm4 and update the fitness
15:   Use Eq.(7) to update the position of antlions
16:   Update the  $Elite$  and  $Fitness_{Elite}$ .
17:    $t = t + 1$ 
18: end while
19: return  $Fitness_{Elite}$ 

```

---

The 1/5 principle was put forward by Rechenberg in the literature [37] in 1986. Its core is that the parameters in the algorithm should be dynamically changed to ensure that the update rate of the population can be maintained at about 20%. In the original antlion algorithm, the scale of the random walk of the ant population is set to the maximum number of iterations, which leads to an increase in the time and space costs of the algorithm. Therefore, this paper combines the 1/5 principle with the random walk behavior of the ant population, in other words, the scale of random walk is set to 1/5 of the maximum number of iterations.

$$\chi(t) = [0, \text{Sum}(2\gamma(t_1) - 1), \text{Sum}(2\gamma(t_2) - 1) \dots, \text{Sum}(2\gamma(t_{\frac{\text{Maxiter}}{5}}) - 1)]. \quad (9)$$

Figure 3 displays the convergence curves of different travel scales in the test function  $F_7(x) = \sum_{i=1}^d ix_i^4 + \text{rand}(0, 1)$ . It can be seen that in the process of 1000 iterations when the scale of random walk is 1/5 of the maximum number of iterations, the optimization accuracy of the algorithm is substantially higher than other ratios.

## 4.2 Piecewise Linear Chaos Strategy

The chaotic search algorithm is a local optimization algorithm with a lot of potential. It can effectively prevent the optimization algorithm from settling for the local optimal solution, and it can converge to the optimal solution faster than the standard random search [38]. The chaotic graphs commonly seen in the literature are as follows:

1. Logistic map[36]:

$$\eta_{n+1} = \mu\eta_n(1 - \eta_n), \mu \in (0, 4] \quad (10)$$

2. Tent map[39]

$$\eta_{n+1} = \alpha \min \{ \eta_n, 1 - \eta_n \}, \alpha \in (0, 2] \quad (11)$$

3. Sine map[40]

$$\eta_{n+1} = -\frac{\gamma}{4} \sin(\pi\eta_n), \gamma \in (0, 4] \quad (12)$$

4. Cubic map[40]

$$\eta_{n+1} = \lambda\eta_n^3 + (1 - \lambda)\eta_n, \lambda \in (0, 4] \quad (13)$$

5. Circle map[39]

$$\eta_{n+1} = \eta_n - \kappa \sin \eta_n, \kappa \in (0, 4.5] \quad (14)$$

6. Piecewise map[40]

$$\eta_{n+1} = \begin{cases} \eta_n & 0 \leq \eta_n \leq \varphi \\ \frac{\eta_n - \varphi}{0.5 - \varphi} & \varphi \leq \eta_n \leq 0.5 \\ \frac{1 - \eta_n - \varphi}{0.5 - \varphi} & 0.5 \leq \eta_n \leq 1 - \varphi \\ \frac{1 - \eta_n}{\varphi} & 1 - \eta_n \leq \eta_n \leq 1 \end{cases}, \varphi \in (0, 0.5] \quad (15)$$

Figure 4 illustrates the bifurcation diagrams of the six classic mappings. Literature [39] pointed out that compared with cosine, logistic and square mapping, tent, sine and circle mapping are most likely to generate more random and uniform pseudo-random number distribution sequences.

Zhenxing et al. [41] proposed an antlion optimization algorithm for adaptive tent chaotic search, which applied tent mapping with a parameter of 0.5 to population initialization and random walk of ants, improving the convergence speed, optimization accuracy and search efficiency of ALO algorithm, as well as its ability to get away from local optimum. However, tent mapping is not perfect. After a certain number of iterations, it tends to the fixed point of tent mapping, and there are small periods and unstable periodic points in the chaotic sequence. Once the data falls into the period of fixed points, it is bound to cause the chaotic sequence to tend to a fixed value and make the algorithm ineffective. Therefore, it is necessary to redesign the sequence generation method in an application, which increases the coding burden. It can be seen from Fig. 5 that the tent map reaches the completely chaotic state only when the parameter is 2, and the piecewise linear can reach the chaotic state within the parameter range.

Figure 5 displays the distribution waveforms of the piecewise linear mapping and tent mapping 500 iterations, where the initial value of chaos is set to  $\chi_0 = 0.68, \alpha = 1.999, \varphi = 0.35$ . Figure 6 demonstrates the distribution of chaotic sequences generated by piecewise linear map and tent map under the same conditions as Fig. 5 in the interval  $[0, 0.1], [0.1, 0.2], \dots [0.9, 1]$ . It is not difficult

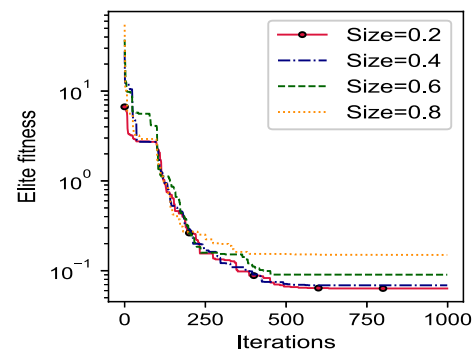
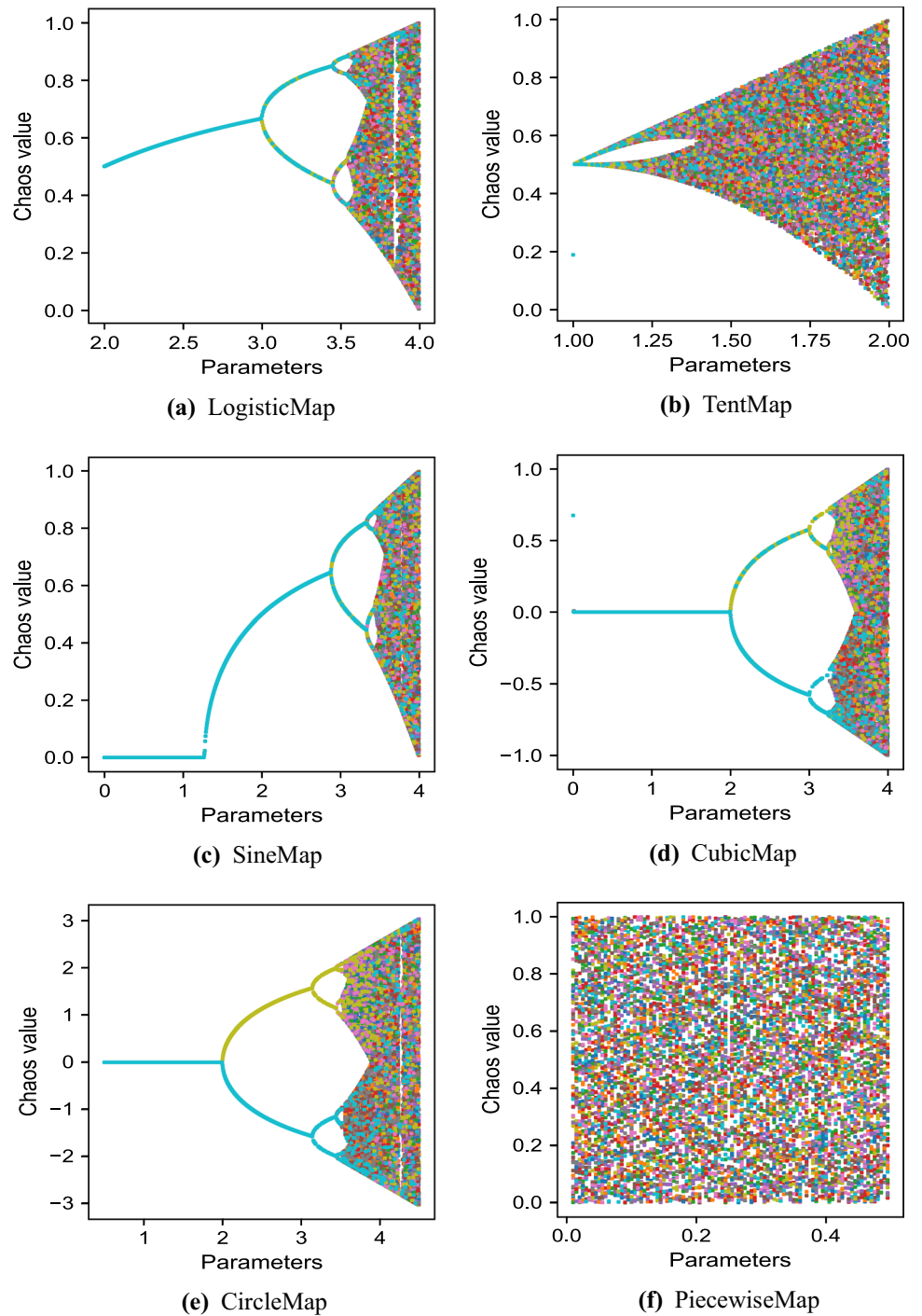
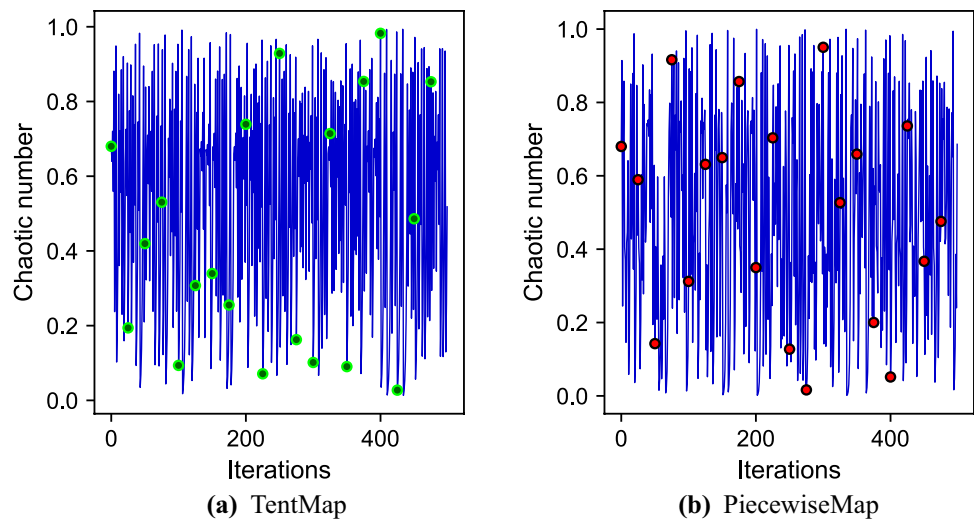
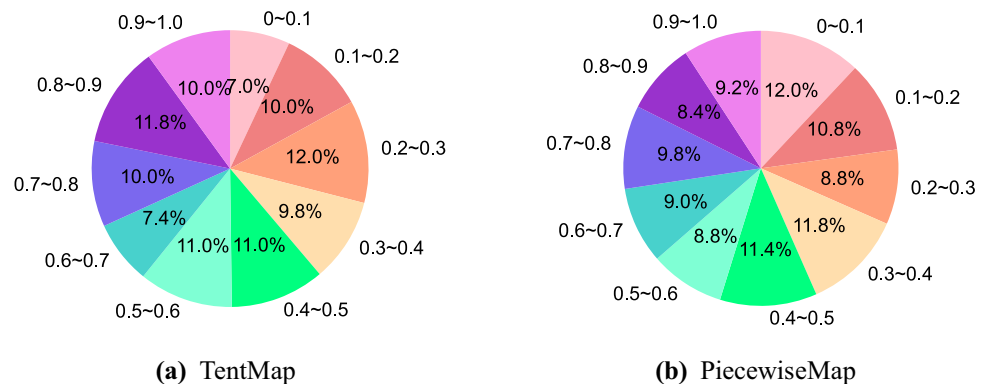


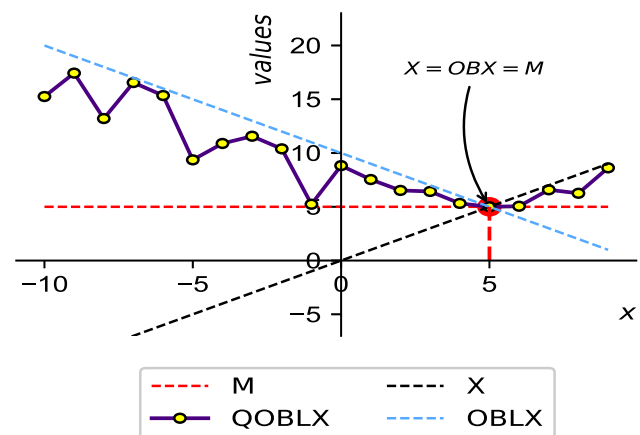
Fig. 3 Convergence curves of different walk sizes in test function F7

**Fig. 4** Bifurcation graphs of six classical mappings

**Fig. 5** Waveforms of piecewise linear and tent mapping**Fig. 6** Numerical distribution of piecewise linear mapping and tent mapping

to see that the piecewise linear mapping can also produce a fairly uniform distribution.

Therefore, this article first uses the chaotic sequence generated by the piecewise linear chaotic map to initialize the population. The specific steps are described in Algorithm 2. In the continuous iteration process of the ALO algorithm, the boundary shrinkage ratio  $I$  increases linearly exponentially, which leads to the decrease of the step length of the ants' random walk and the decrease of the diversity of the antlion population, which easily causes the algorithm to become trapped in the local optimal and cannot jump out. Therefore, this paper combines the piecewise linear chaotic mapping with the random walk operation of the ant population to improve the exploration and development capabilities of the ant population and the ability of the algorithm to escape from the local optimal solution. The algorithm is marked as Algorithm 3, and the specific steps are as follows:

**Fig. 7** Opposite number and quasi-opposite number



**Algorithm 2** Initialize the population

**Input:** Set the parameters. Including group size  $N$ , individual dimensions  $d$ , parameters  $\phi$ , upper boundary  $ub$ , lower boundary  $lb$ , population location  $Population$  and corresponding population fitness  $Fitness$ .

**Output:** the position and fitness of  $Population$

- 1: Randomly generates  $d$  initial values  $\chi_{0j}, (j = 1, 2, \dots, d)$  in  $[lb_j, ub_j]$ , and then use Eq.(15) to generate  $d$  chaotic sequences of length  $N$  to obtain a chaotic array  $Piecewise_{chaos}(N \times d)$

$$Population[i, j] = Piecewise[i, j] \times (ub_j - lb_j) + lb_j, (i = 1, 2, \dots, N) \quad (16)$$

- 2: Map the value of the chaotic array into the search space according to Eq.(16), and assign it to the population position accordingly, and then find the corresponding fitness value.
- 3: **return**  $Population$

**Algorithm 3** Chaotic random walk operation of the ant population

**Input:** Set parameters. It includes the step size of random walk  $M$ , individual dimension  $d$ , upper bound  $ub$ , lower bound  $lb$ , parameters  $\phi$  and the position of ant population  $RW(M \times d)$  after the random walk.

**Output:** The random walk  $RW$

- 1: Randomly generates  $d$  initial value  $\chi_{0j}, (j = 1, 2, \dots, d)$  in  $[lb_j, ub_j]$  and then uses the Eq.(15) to generate  $d$  chaotic sequences of length  $M$  to obtain a chaotic array  $Piecewise_{chaos}(M \times d)$ .

$$RW[i, j] = Piecewise[i, j] \times (ub_j - lb_j) + lb_j, (i = 1, 2, \dots, N) \quad (17)$$

- 2: map the value of the chaotic array into the search space according to Eq.(17)
- 3: **return**  $RW$

### 4.3 Quasi-Opposite Learning Strategy

The notion of Opposition-Based Learning (OBL) was first presented by Tizhoosh [42]. OBL is utilized in the field of computational intelligence to speed up the convergence speed of various optimization approaches.

If is a point  $\chi(\chi_1, \chi_2, \dots, \chi_d), \{\chi_1, \chi_2, \dots, \chi_d\} \in R, \chi_j \in [\chi_j^{\min}, \chi_j^{\max}]$  in the  $d$  dimensional search space, the quasi-opposite point  $QO\chi(QO\chi_1, QO\chi_2, \dots, QO\chi_d)$  of  $\chi$  is expressed as:

$$QO\chi_0 = \text{rand}\left(\frac{\chi_j^{\max} + \chi_j^{\min}}{2}, \chi_j^{\max} + \chi_j^{\min} - \chi_j\right), (j = 1, 2, \dots, d) \quad (18)$$

Figure 7 shows the number of oppositions and quasi-oppositions of  $X = [1, 2, \dots, 10]$  when the search space boundary is  $[lb, ub]$ . It can be seen that when  $M > X$ ,

QOBLX is located between  $M$  and OBLX ;when  $M < X$ , QOBLX is located between  $M$  and  $X$ , and when  $M = X$ , QOBLX = OBLX.

At the initial stage of the algorithm, ants and antlions with poor fitness could be generated, resulting in elite individuals with poor fitness, and further lead to the whole ant population learning from antlions of poor fitness, thus delaying the optimization process of the algorithm. Therefore, this paper uses the quasi-opposite strategy to optimize the initial population after initialization, so that the randomly generated initial position and its quasi-opposite position can be considered at the same time. By expanding the search space by exploring a larger region, the quality of the initial population can be improved and the algorithm can be accelerated. Besides, in the iterative process, the quasi-opposite learning strategy is used to optimize the ant population, which can accelerate the speed of the ant searching in the solving space. The algorithm is named as Algorithm 4.

**Table 1** Multi-dimensional single-peak test function

Function name	Dimensions	Variable range	Best value
$F_1(x) = \sum_{i=1}^d x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^d \left( \sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max\{ x_i \}, 1 \leq i \leq d$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^d (x_i + 0.5)^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^d ix_i^4 + \text{rand}(0, 1)$	30	$[-1.28, 1.28]$	0

**Table 2** Multi-dimensional multi-peak test function

Function name	Dimensions	Variable range	Best value
$F_8(x) = \sum_{i=1}^d -x_i \sin(\sqrt{ x_i })$	30	$[-100, 100]$	0
$F_9(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$[-10, 10]$	0
$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{\sum_{i=1}^d x_i^2}{d}}) + 20 + e - \exp(\sqrt{\frac{\sum_{i=1}^d \cos(2\pi x_i)}{d}})$	30	$[-100, 100]$	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \frac{x_i}{\sqrt{i}} + 1$	30	$[-100, 100]$	0

**Table 3** Experimental results on multidimensional unimodal functions based on different algorithms

Function	Metrics	ALO	DA	SCA	IALOT	CAA
F1	Mean	0.0000	1373.9167	0.0359	0.0000	0.0000
	SD	0.0000	289486.1745	0.0112	0.0000	0.0000
	Best	0.0000	312.5000	0.0000	0.0000	0.0000
	Worst	0.0001	2365.3000	0.5228	0.0000	0.0000
F2	Mean	56.9928	14.2444	0.0000	0.0001	0.0000
	SD	47.8273	41.0578	0.0000	0.0000	0.0000
	Best	0.0393	4.0381	0.0000	0.0001	0.0000
	Worst	129.6563	33.7598	0.0004	0.0001	0.0000
F3	Mean	1166.4913	13456.4667	4610.4000	0.0000	0.0000
	SD	663.7232	66641538.2600	18650756.8700	0.0000	0.0000
	Best	263.7008	1507.0000	204.0000	0.0000	0.0000
	Worst	3235.0773	34737.0000	16417.0000	0.0000	0.0000
F4	Mean	13.5164	22.7198	19.1240	0.0001	0.0000
	SD	4.1912	34.5281	81.0951	0.0000	0.0000
	Best	7.1046	14.1493	4.4369	0.0001	0.0000
	Worst	24.4321	35.9957	45.2264	0.0001	0.0000
F5	Mean	121.6281	131088.0000	930.8733	28.4348	1.4082
	SD	262.5982	24664273892.0000	3555169.7650	0.0383	1.6687
	Best	14.1400	4580.0000	28.2000	28.3657	0.0000
	Worst	1424.3866	752820.0000	8370.0000	28.4897	5.3587
F6	Mean	0.0000	1168.8200	4.6942	0.0000	0.2520
	SD	0.0000	611869.1010	0.4572	0.0000	0.2162
	Best	0.0000	135.3000	3.6818	0.0000	0.0000
	Worst	0.0000	3996.4000	6.2725	0.0000	0.7116
F7	Mean	0.0979	0.3118	0.0364	0.0000	0.0000
	SD	0.0406	0.0205	0.0011	0.0000	0.0000
	Best	0.0396	0.1054	0.0042	0.0000	0.0000
	Worst	0.2004	0.5950	0.1247	0.0002	0.0001

**Algorithm 4** Optimize population location

**Input:** Need to set parameters. Including group size  $N$ , individual dimensions  $d$ , group location  $Population$  and corresponding population fitness  $Fitness$

**Output:** Return to the updated population  $Population$  with better fitness and the corresponding fitness  $Fitness$ .

- 1: Find the position of each individual in the population according to Eq.(18) to obtain the quasi-opposite population  $QOPopulation$
- 2: Find the corresponding population fitness according to the obtained quasi-opposite population  $QOPopulation$ . Use the better-adapted individuals in the quasi-opposite population to replace the less-adapted individuals in the original population.
- 3: **return**  $Population, Fitness$

## 4.4 Experiments on Classic Benchmark Functions

### 4.4.1 Test Function and Parameter Settings

To verify the effect of CAA, the functions are tested with 11 different feature standards, and simulation experiments are carried out in multiple dimensions. All the experiments are carried out with *PyCharmCommunityEdition2020.3.1* and *MATLABR2018b* on a *Windows* platform using *AMD-Ryzen74800U* with *RadeonGraphics1.8 GHz* computer. Besides, CAA is compared with the original ALO algorithm [7], Dragonfly algorithm (DA) [43], Sine Cosine Algorithm (SCA) [44] and IALOT [45] algorithm for function optimization results. Table 1 shows the basic parameters and characteristics of the multi-dimensional unimodal test function. Table 2 displays the basic parameters and characteristics

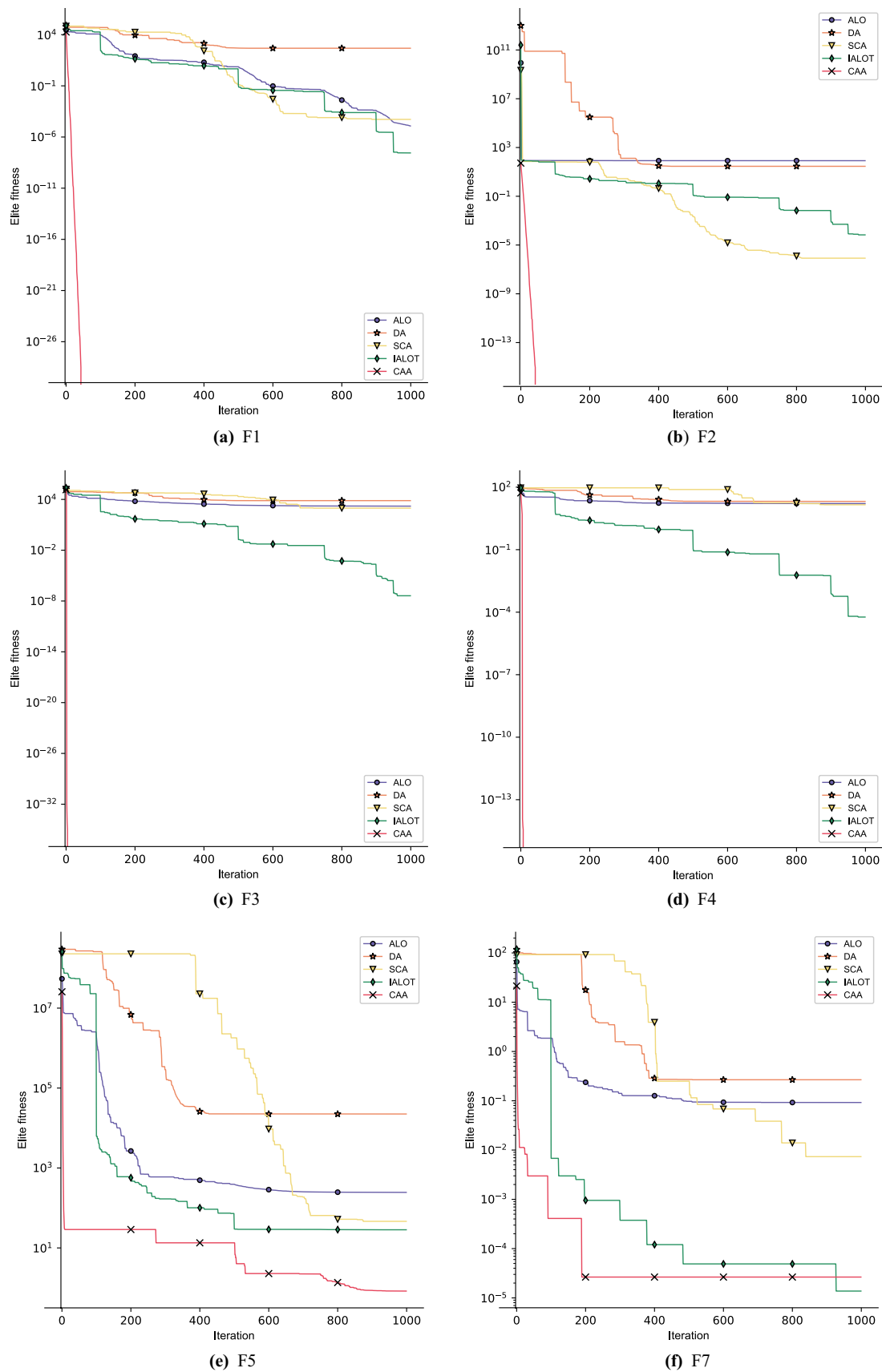
of the multi-dimensional and multi-peak test function. The multi-dimensional single-peak function in the test function is utilized evaluate the optimization accuracy and convergence speed of the algorithm, and the multi-dimensional multi-peak function is used to evaluate the ability of the algorithm to leap out of the local optimal solution and the global convergence performance.

### 4.4.2 Experimental Results and Analysis

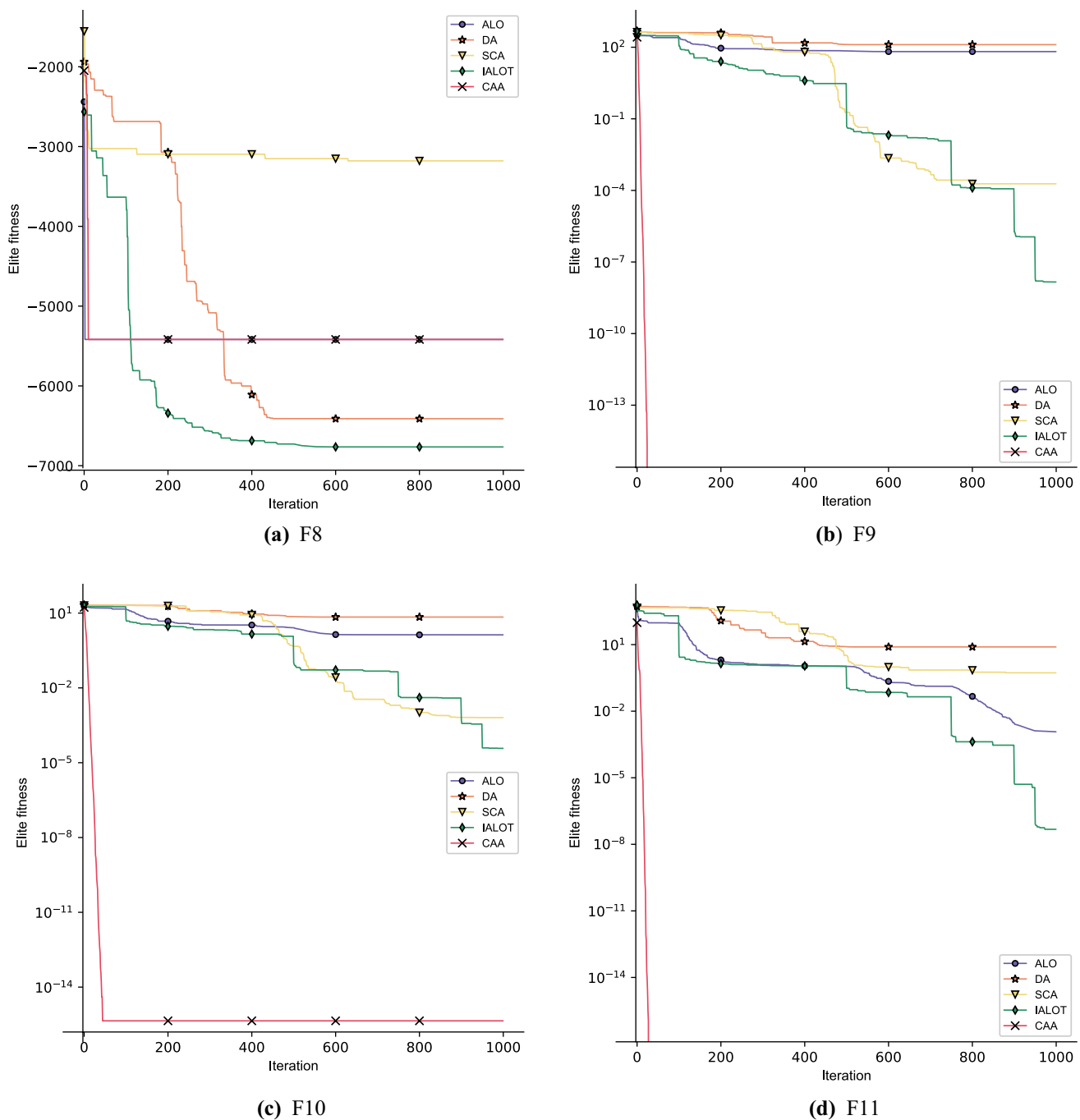
To guarantee that the experiment is completely fair, the above five algorithms set unified parameters: the population size is 30, the variable dimension is 30, the maximum number of iterations is 1000, and the parameter of PWLCM is  $\varphi = 0.35$ . Taking into account the randomness of the algorithm, each test function is executed 30 times. The optimal

**Table 4** Experimental results on multidimensional multimodal functions based on different algorithms

Function	Metrics	ALO	DA	SCA	IALOT	CAA
F8	Mean	− 5771.5058	− 5499.3733	− 3939.0100	− 7107.6353	− 6094.9302
	SD	1097.5791	417811.5937	94693.5830	565.5526	1115.0020
	Best	− 11036.3348	− 7063.1000	− 4610.8000	− 8364.7245	− 9368.0764
	Worst	− 5417.6748	− 4092.6000	− 3454.5000	− 5501.9949	− 4642.4685
F9	Mean	94.6535	176.6170	17.5051	0.0000	0.0000
	SD	27.7241	1404.5384	403.1920	0.0000	0.0000
	Best	54.7226	77.0817	0.0000	0.0000	0.0000
	Worst	165.1623	252.1318	73.2185	0.0000	0.0000
F10	Mean	2.1428	8.3607	12.5365	0.0000	0.0000
	SD	0.4760	3.1986	85.9052	0.0000	0.0000
	Best	1.1231	5.0280	0.0002	0.0000	0.0000
	Worst	3.0269	11.5911	20.3115	0.0000	0.0000
F11	Mean	0.0152	13.5666	0.2423	4.37−08	0.0000
	SD	0.0163	34.3603	0.0612	0.0000	0.0000
	Best	0.0002	2.5745	0.0000	0.0000	0.0000
	Worst	0.0594	28.8537	0.6954	0.0000	0.0000



**Fig. 8** Convergence curve of five algorithms on multidimensional unimodal functions



**Fig. 9** Convergence curve of five algorithms on multidimensional multimodal functions

solution, worst solution, mean, and variance obtained by the five algorithms on the test function are used as evaluation indicators. The experimental findings of the multi-dimensional unimodal function are displayed in Table 3, and the experimental findings of the multi-dimensional multimodal function are shown in Table 4.

According to the experimental results in Table 3, It can be seen that for the multidimensional single-peaked functions

F1, F2, F3, and F4, CAA can obtain the optimal values of the functions, and the optimization success rate is as high as 100%. However, although the optimal and average values obtained on F5 and F7 do not reach the theoretical optimal values, the results are more accurate and stable than those of other algorithms. This indicates that CAA has good optimization capability for multidimensional unimodal functions. Not only the optimization accuracy is high but also



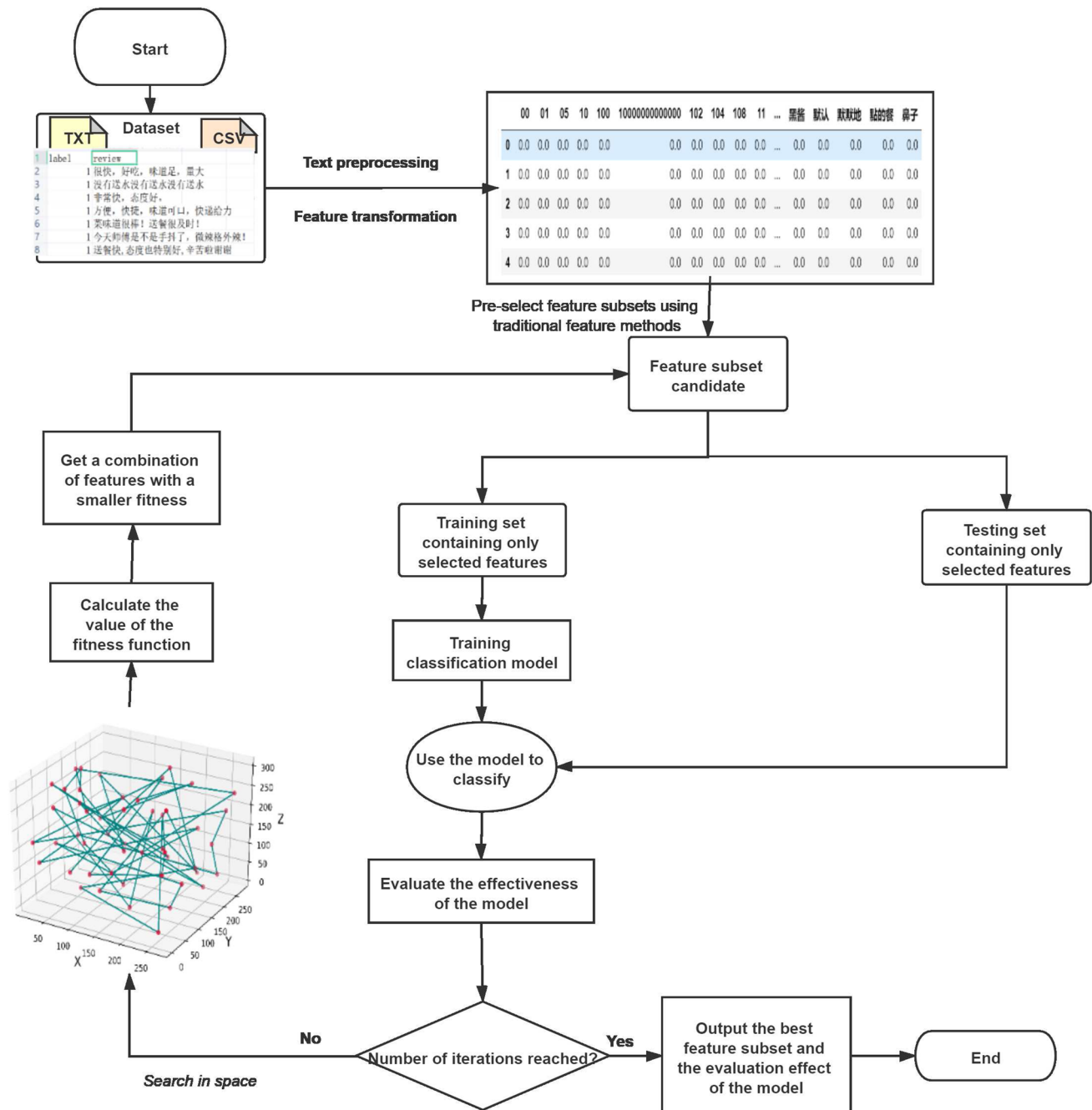
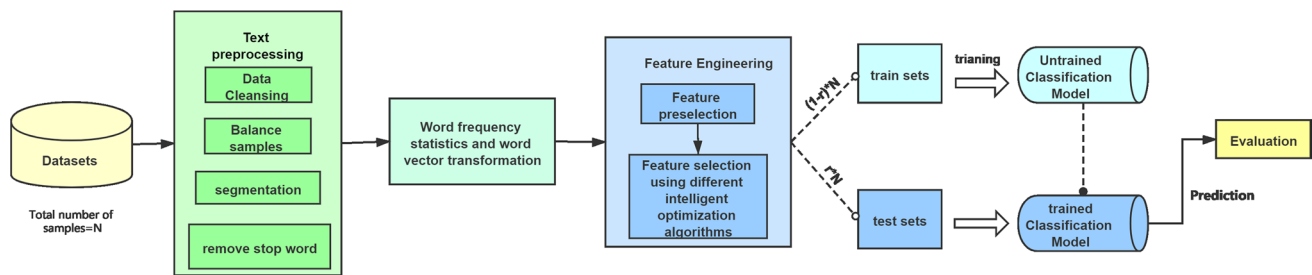


Fig. 10 Steps of Chinese text classification based on CAFSA algorithm

the robustness is strong. The experimental results in Table 4 demonstrate that for the multi-dimensional single-peak function  $F8$ , the average value obtained by CAA is not as good as that of IALOT, but the optimal value produced is better than that of IALOT; For function  $F9$ , the average value produced by CAA reaches  $4.44E - 16$ , which is markedly better than the original ALO algorithm DA, SCA and IALOT. For functions  $F10$  and  $F11$ , it can be noticed that the mean value is 0, indicating that CAA can find the theoretical optimum,

and the variance is 0 indicating that the optimal solution was found in each of the 30 randomized independent experiments, which is significantly outperformed by the ALO algorithm, DA and SCA algorithms. To compare the optimization performance and convergence effects of CAA and the original ALO algorithm, DA, SCA and IALOT algorithm more intuitively, Figs. 8 and 9 are the convergence curves of 9 test functions in one of 30 independent experiments. Figure 8 illustrate that as the number of iterations increases,



**Fig. 11** General framework of Chinese text classification task

CAA has a more prominent advantage than other algorithms in terms of optimization speed. Besides, the optimization accuracy of CAFSA method is considerably greater than other algorithms when the number of iterations is about 20 generations, as shown in Fig. 8c, d. The above reflects the efficient optimization ability of CAA in multi-dimensional single-peak functions. Although the solution accuracy of CAA in Figs. 8f and 9a is not optimal, the convergence speed is faster than other algorithms. It can be seen from Fig. 9b–d that as the iterative process continues, whether it is the optimization accuracy or the convergence speed of the algorithm, the effect of applying CAFSA is obviously more efficient than other algorithms. The analysis of the experimental results indicates that the improved antlion optimization algorithm has fast convergence speed and high optimization accuracy, and is effective and reliable in function optimization.

## 5 The Chaotic Antlion Feature Selection Algorithm (CAFSA) Using CAA

Text classification is a pre-defined text category, which is then divided into one or more topic categories according to the content of the text to be classified. Dividing text documents into corresponding categories is helpful for users to retrieve and recognize required information on web pages or huge corpora. The traditional Chinese text classification process mainly includes text corpus acquisition, text preprocessing, text word vector representation, construction of a classification model, and evaluation of test text. In the process of Chinese text classification, a document usually contains thousands of unique words, and each word is regarded as a feature. But these features may contain a small amount of information or have nothing to do with the category label. At present, the text features selection methods used in academic circles are mainly chi-square test (CHI), F test, information gain (IG), and other methods. These traditional feature selection methods are to calculate the importance of each feature to the classification through the evaluation function, and then select those features that meet the quantitative

requirements and have a large degree of importance as part of the feature subset. Since the interaction between features is complex, some features in the selected feature subset may not guarantee the best classification effect. Therefore, this paper proposes a new feature selection mechanism for Chinese text classification. On the basis of preselecting the original features through the evaluation function, CAFSA is used to further select more representative features, so as to consider improving the classification accuracy of the model while reducing the number of features. Figure 10 is the specific process of feature selection using CAFSA. Figure 11 shows the framework for feature selection using intelligent optimization algorithms in Chinese text classification tasks.

1. The corpus or text can be obtained through web crawlers or download public data sets
2. Perform data preprocessing. The preprocessing process of this experiment consists of four steps. First perform data cleaning, delete duplicate or missing data, and remove some unnecessary characters from the text through regular expressions. Then use down-sampling technology to process the unbalanced data set so that the samples of each category are basically the same. Then this thesis uses jieba word segmentation technology to segment the text. Finally, load the stop word list and delete invalid stop words in the text after word segmentation.
3. Word frequency statistics and text word vector representation. This paper uses TFIDF technology to count word frequency.

**Table 5** Parameter settings for different optimization methods

Parameter	Value	Optimizer
Maximum iterations	50	ALO, CALOP, CAFSA, SOA, HHO, SSA
Population size	10	ALO, CALOP, CAFSA, SOA, HHO, SSA
Chaotic parameter $\varphi$	0.35	CAFSA, CALOP

4. Feature selection. This experiment first uses Chi-square test for feature pre-selection and then the aforementioned intelligent optimization algorithm for optimization.
5. The division of training set and test set. Use the selected feature subset to update the data set, and divided the training and the test set by  $(1 - r) : r$ ,  $r$  was taken as 0.3.
6. Classification model. This experiment uses a support vector machine classifier as a classification tool. First, the classification model is trained with the training data. The trained classifier is then applied to forecast the categories of the test set and finally the classification performance is assessed.

## 6 Experimental Procedure and Analysis of Results

This article uses PyCharm Community Edition 2020.3.1 for simulation under Windows 10 64-bit operating system, 16G running memory, AMD Ryzen 7 4800U with Radeon Graphics 1.8 GHz environment. The research object of this algorithm includes three different data sets. In addition, the effectiveness of CAFSA feature selection algorithm is verified by comparing with the original antlion optimization algorithm (ALO) [7], Piecewise linear chaotic antlion optimization (CALOP) [36], Seagull Optimization Algorithm (SOA) [46], Sparrow Search Algorithm (SSA) [47], and Harry Hawk Optimizer [48]. Feature selection techniques.

### 6.1 Preparation of Data Set and Setting of Parameters

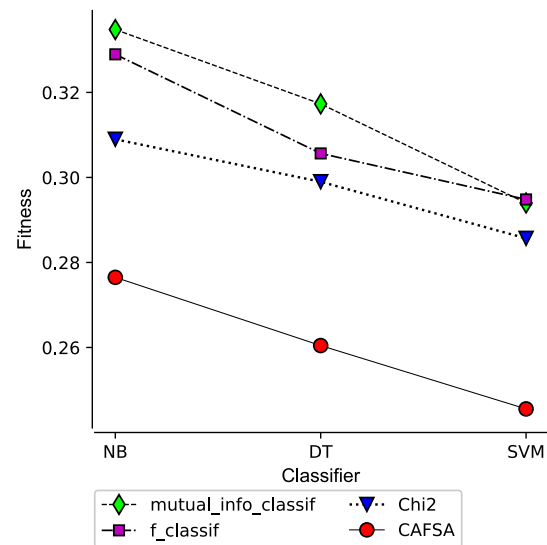
In the experiment, three different data sets were used in this research work. For all data sets, we use the same preprocessing techniques and experimental procedures shown in Fig. 11.

1. SogouCS: Collected Sohu news data from 18 channels from June to July 2012, including domestic, sports, international and entertainment. In addition, website and text information are also provided. The nine categories with the most samples in total were selected for the experiment, with 300 samples per category, for a total of 2700 data. In particular, the data set was separated into a training set and a test set according to a 2:1 ratio.
2. ChnSenticorp\_Hotel, ChnSenticorp\_NoteBook: From two Chinese sentiment mining corpora collected and collated by Tan Songbo, each containing 2000 positive and negative comment texts.

The parameters involved in the wrapper feature selection method used in the experiment are shown in Table 5. In

**Table 6** Classic feature selection method and CAFSA experimental results on SoGouCS news data set

Classifier method	SVM	NB	DT
CAFSA	0.24554630	0.27649773	0.26045111
Chi2	0.28566142	0.30897317	0.29898242
<i>f_classif</i>	0.294819611	0.328954672	0.305642923
<i>mutual_info_classif</i>	0.293987049	0.334782609	0.317298797



**Fig. 12** Chi-square test and CAFSA performance on SoGouCS news data set

addition, choose three classic filter feature selection methods, as follows:

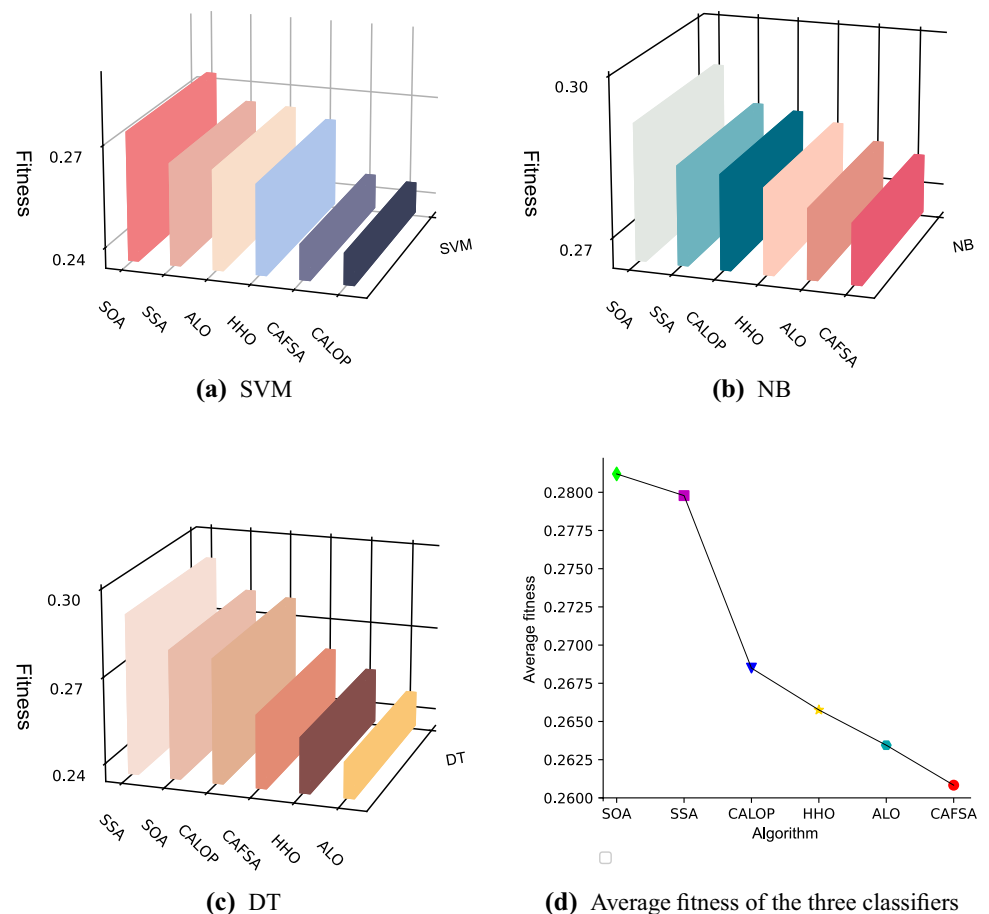
1. *f\_classif*: ANOVA  $F$  value between label/feature for classification tasks.
2. *mutual\_info\_classif*: Mutual information for a discrete target.
3. *Chi2*: Chi-squared stats of non-negative features for classification tasks.

### 6.2 Evaluation Criteria

In the experiment, support vector machine (SVM) and naive Bayes (NB) decision tree (DT) classifiers were selected to evaluate the classification performance of each algorithm, and the radial basis function was selected as the kernel function. In addition, the goal of the experiment in this section is to use intelligent optimization algorithms to improve the accuracy of model classification while reducing the number of features. The fitness function is given in Eq. (19)

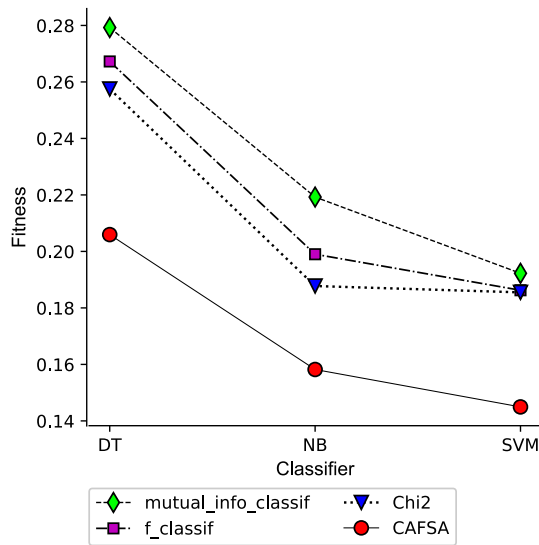
**Table 7** Experimental results of six intelligent algorithms on the SoGouCS news data set

Classifier	Criterion method	ASize	Precision	Macro-F1 score	MFitness	Rank
SVM	ALO	0.506	0.762257	0.775376	0.264519	4
	CALOP	0.513	0.785384	0.713948	0.244404	1
	CAFSa	0.616	0.795560	0.803620	0.245546	2
	SOA	0.520	0.754329	0.737073	0.273104	6
	HHO	0.669	0.783534	0.790847	0.261670	3
	SSA	0.445	0.754857	0.766939	0.265129	5
NB	ALO	0.552	0.752081	0.734514	0.278327	2
	CALOP	0.507	0.741906	0.651829	0.282935	4
	CAFSa	0.491	0.747280	0.716051	0.276498	1
	SOA	0.495	0.731795	0.711062	0.290835	6
	HHO	0.490	0.741906	0.724010	0.281285	3
	SSA	0.348	0.723404	0.703304	0.283736	5
DT	ALO	0.494	0.779833	0.788539	0.247500	1
	CALOP	0.509	0.747456	0.679896	0.278140	4
	CAFSa	0.457	0.761332	0.771595	0.260451	3
	SOA	0.491	0.743756	0.742919	0.279670	5
	HHO	0.379	0.759482	0.774805	0.254316	2
	SSA	0.391	0.720629	0.735427	0.290484	6

**Fig. 13** Comparison of different feature selection algorithms on SoGouCS dataset

**Table 8** Classic feature selection method and CAFSA experimental results on ChnSenticorp\_Hotel dataset

Classifier method	SVM	NB	DT
CAFSA	0.14495	0.1582	0.20595
Chi2	0.1855	0.18775	0.2575
f_classif	0.18625	0.199	0.26725
mutual_info_classif	0.19225	0.21925	0.27925

**Fig. 14** Chi-square test and CAFSA performance on ChnSenticorp\_Hotel data set

$$\text{Fitness}_i = \alpha \left( 1 - \frac{\text{Num}_c}{\text{Num}_t} \right) + (1 - \alpha) \left( \frac{\text{Count}_f}{\text{Count}_z} \right), \quad (19)$$

where  $\text{Num}_c$  is the number of samples that were properly categorized,  $\text{Num}_t$  is the total number of samples in the data set. Where  $\text{Count}_f$  is the number of features selected by the optimization algorithm,  $\text{Count}_z$  is the total number of features in the data set.  $\alpha$  represents the weight, which is 0.9 here.

The following evaluation metrics were applied to assess the capabilities of the algorithm in different aspects:

1. *Average feature size* It's the proportion of the algorithm's chosen features to the overall number of features in the data set. It is also one of the goals considered by the fitness function. It can be expressed by Eq. 20:

$$\text{ASize} = \frac{1}{M} \sum_{i=1}^M \frac{\text{size}(D_i)}{T} \quad (20)$$

Among them,  $M$  is the number of independent runs of the experiment, taking 10.  $\text{size}(D_i)$  is the number of features selected by the algorithm for the  $i$ -th execution

of the data set  $D$ , and  $T$  is the total number of features of the data set  $D$ .

2. *Average classification precision*

$$\text{Precision} = \frac{1}{M} \sum_{i=1}^M \text{accuracy}(D_i), \quad (21)$$

where  $\text{accuracy}(D_i) = \frac{\text{Num}_c}{\text{Num}_t}$

3. *Average macro-f1 score*

$$\text{Macro-F1 score} = \frac{1}{M} \sum_{i=1}^M f\text{score}(D_i). \quad (22)$$

Among them,  $f\text{score}(D_i)$  is the score obtained when the algorithm is executed for the  $i$ -th time on the data set  $D$ .  $= (1 + \beta^2) * \frac{\text{Precision} * \text{Recall}}{(\beta^2 * \text{Precision}) + \text{Recall}}$ .  $\beta$  is the reconciliation parameter, which is 1 here. Precision is the accuracy rate and Recall is the recall rate.

4. *Mean Fitness*

$$\text{MFitness} = \frac{1}{M} \sum_{i=1}^M \text{Fitness}_i \quad (23)$$

## 6.3 Results and Analysis

### 6.3.1 Results on SoGouCS Dataset

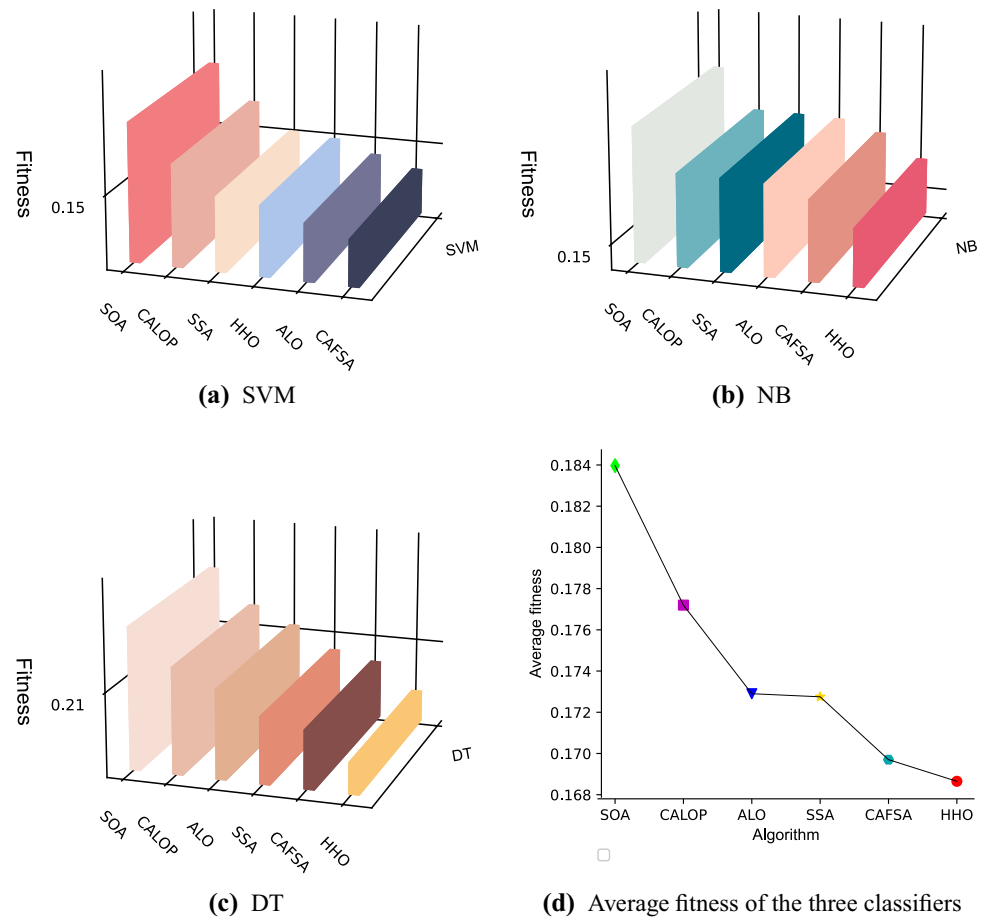
Table 6 shows the experimental results of using traditional feature selection methods such as chi-square test to directly preselect 2000-dimensional features on the Sogou news data set, and using CAFSA algorithm for further optimization, and then using support vector machines, naive Bayes and decision-making Tree classifier for classification. The visualization of Table 6 is shown in Fig. 12. It can be seen that CAFSA improves the classification result after feature selection. Specifically, when using the SVM classifier, the fitness function is reduced by 16.337092% compared with the chi-square test optimized using CAFSA. In contrast, when the naive Bayes classifier and the decision tree classifier are used, the drops are 11.7452823% and 14.7940677%, respectively.

Table 7 is the result of feature selection using six different feature selection methods after using traditional chi-square test for feature preselection. To show the comparison effect more intuitively, Fig. 13 shows the fitness comparison under three classifiers and six feature selection methods.

As can be seen from Fig. 13, CAFSA performs well on the SoGouCS dataset when using the Support Vector Machine, Parsimonious Bayes and Decision Tree classifiers respectively. In particular, CAFSA achieves a better fitness of 0.276498 than the other algorithms when using the Naive Bayes classifier, and a fitness of 0.245546 when using the



**Fig. 15** Comparison of different feature selection algorithms on ChnSenticorp\_Hotel dataset

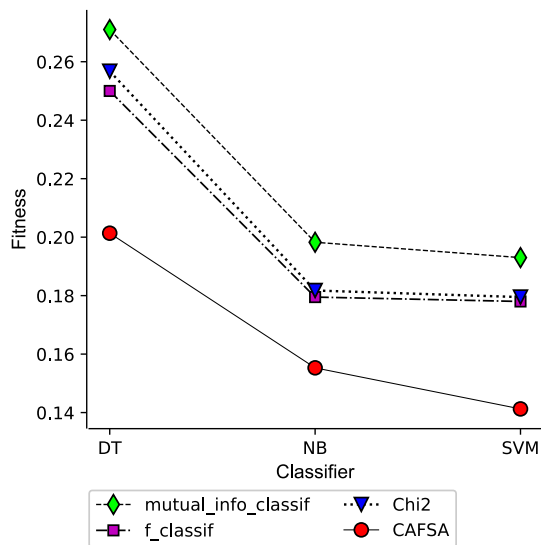


**Table 9** Experimental results of six intelligent algorithms on the ChnSenticorp\_Hotel data set

Classifier	Criterion method	ASize	Precision	Macro-F1 score	MFitness	Rank
SVM	ALO	0.491	0.890833	0.894113	0.147350	2
	CALOP	0.473	0.878333	0.878138	0.156800	5
	CAFSA	0.497	0.894167	0.890751	0.144950	1
	SOA	0.514	0.874167	0.874103	0.164650	6
	HHO	0.473	0.885833	0.885798	0.150050	3
	SSA	0.354	0.871667	0.871597	0.150900	4
NB	ALO	0.472	0.875000	0.874972	0.159700	3
	CALOP	0.504	0.878333	0.878332	0.159900	5
	CAFSA	0.472	0.876667	0.876664	0.158200	2
	SOA	0.487	0.870000	0.849996	0.165700	6
	HHO	0.387	0.870833	0.870736	0.154950	1
	SSA	0.481	0.875833	0.875826	0.159850	4
DT	ALO	0.474	0.817500	0.816963	0.211650	4
	CALOP	0.499	0.816667	0.816581	0.214900	5
	CAFSA	0.447	0.820833	0.820273	0.205950	2
	SOA	0.468	0.805833	0.805735	0.221550	6
	HHO	0.427	0.824167	0.824102	0.200950	1
	SSA	0.485	0.823333	0.823237	0.207500	3

**Table 10** Classic feature selection method and CAFSA experimental results on ChnSenticorp\_NoteBook dataset

Classifier method	SVM	NB	DT
CAFSA	0.14125	0.1553	0.20135
Chi2	0.1795	0.18175	0.25675
<i>f_classif</i>	0.178	0.1795	0.25
<i>mutual_info_classif</i>	0.193	0.19825	0.271

**Fig. 16** Chi-square test and CAFSA performance on ChnSenticorp\_NoteBook data set

support vector machine classifier, second only to the CALOP algorithm at 0.244404. When using the decision tree classifier, CAFSA is second only to the ALO algorithm and the HHO algorithm. Figure 13d plots the average of the fitness obtained by the six algorithms using the three classifiers, which shows that CAFSA outperforms the other five intelligent algorithms on average for the three different classifiers.

### 6.3.2 Results on ChnSenticorp\_Hotel Dataset

Table 8 shows the experimental results of using traditional feature selection methods such as chi-square test to directly preselect 1000-dimensional features on ChnSenticorp\_Hotel data set, and using CAFSA algorithm for further optimization, and then using support vector machines, naive Bayes and decision-making Tree classifier for classification. Figure 14 is a visual display of Table 8. It can be seen that after the feature selection by CAFSA, the classification effect has been improved. Specifically, when the SVM classifier is used, the fitness function optimized by CAFSA is reduced by 21.85983827% compared with the chi-square test, and when the naive Bayes and decision tree classifiers are used,

they are reduced by 15.73901465% and 20.01941748%, respectively.

Table 9 is the result of feature selection using six different feature selection methods after using traditional chi-square test for feature preselection. To display the comparison effect more intuitively, Fig. 15 shows the fitness comparison under the three classifiers and six feature optimization methods.

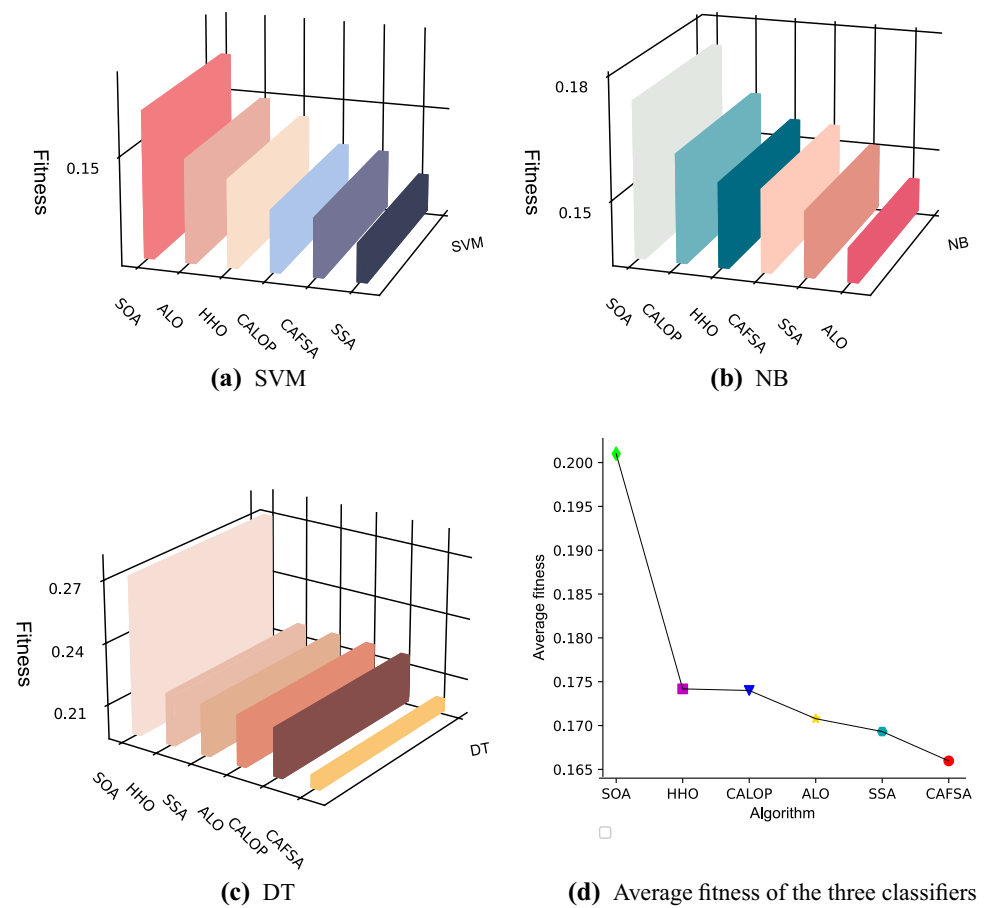
It can be seen from Fig. 15 that CAFSA has excellent performance when using support vector machines, naive Bayes and decision tree classifiers on the ChnSenticorp\_Hotel data set. Especially when using the support vector machine classifier, CAFSA has a better fitness of 0.144950 than other algorithms. When using the Naive Bayes classifier, the fitness of CAFSA is 0.205950, which is second only to 0.200950 of the HHO algorithm. When using a decision tree classifier, CAFSA is second only to the HHO algorithm. Figure 15d presents the average of the fitness obtained by the six algorithms using the three classifiers, from which it can be seen that CAFSA outperforms the SSA, SOA, ALO, and CALOP algorithms on average for the three different classifiers.

### 6.3.3 Results on ChnSenticorp\_NoteBook dataset

Table 10 shows the experimental results of using traditional feature selection methods such as chi-square test to directly preselect 1000-dimensional features on ChnSenticorp\_NoteBook data set, and using CAFSA algorithm for further optimization, and then using support vector machines, naive Bayes and decision-making Tree classifier for classification. Figure 16 is a visual display of Table 10. It can be seen that after the feature selection by CAFSA, the classification effect has been improved. Specifically, when the decision tree classifier is used, the fitness function optimized by CAFSA is reduced by 21.57740993% compared with the chi-square test, while using the SVM classifier and the naive Bayes classifier, it decreased by 21.3091922% and 14.55295736%, respectively.

Table 11 is the result of feature selection on the ChnSenticorp\_NoteBook dataset using CAFSA and the other five different feature selection methods after using the traditional chi-square test for feature preselection. To display the comparison effect more intuitively, Fig. 17 shows the fitness comparison under the three classifiers and six feature optimization methods. It can be seen from Fig. 17 that CAFSA has excellent performance when using support vector machines, naive Bayes and decision tree classifiers on the ChnSenticorp\_NoteBook data set. Especially when using the decision tree classifier, CAFSA has a better fitness of 0.20135 than other algorithms. When using a support vector machine classifier, the fitness of CAFSA is second only to the SSA algorithm. When using the Naive Bayes classifier, CAFSA is second only to the ALO and SSA algorithms.

**Fig. 17** Comparison of different feature selection algorithms on ChnSenticorp\_NoteBook dataset



**Table 11** Experimental results of six intelligent algorithms on the ChnSenticorp\_NoteBook data set

Classifier	Criterion method	ASize	Precision	Macro-F1 score	MFitness	Rank
SVM	ALO	0.53	0.8925	0.892305477	0.14975	5
	CALOP	0.481	0.895833333	0.895507593	0.14185	3
	CAFSA	0.565	0.905833333	0.905712265	0.14125	2
	SOA	0.516	0.881666667	0.881444045	0.1581	6
	HHO	0.553	0.898333333	0.898156522	0.1468	4
	SSA	0.361	0.8875	0.887327157	0.13735	1
NB	ALO	0.518	0.898333333	0.898827755	0.1433	1
	CALOP	0.522	0.878333333	0.878327926	0.1617	5
	CAFSA	0.488	0.881666667	0.881663708	0.1553	3
	SOA	0.52	0.865	0.8649985	0.1735	6
	HHO	0.628	0.896666667	0.8966526	0.1558	4
	SSA	0.468	0.884166667	0.884162725	0.15105	2
DT	ALO	0.498	0.811666667	0.811666144	0.2193	3
	CALOP	0.482	0.810833333	0.810832151	0.21845	2
	CAFSA	0.416	0.8225	0.822485084	0.20135	1
	SOA	0.524	0.756666667	0.756655851	0.2714	6
	HHO	0.437	0.804166667	0.804163267	0.21995	5
	SSA	0.313	0.790833333	0.789093206	0.21955	4

Figure 17d was the mean of the fitness obtained by the six algorithms using three classifiers, from which it can be seen that CAFSA outperforms the other five algorithms on average on three different classifiers.

## 7 Conclusion

This paper proposes a chaotic antlion feature selection algorithm (CAFSA) based on quasi-opposite learning mechanism and chaotic mapping, trying to use it as a feature selection method for Chinese text classification. Firstly, the Piecewise Linear Chaotic Map (PWLCM) is used to initialize the ant and antlion population, and then the original ant and antlion population is optimized using the quasi-opposition learning technique, which enhances the population's variety and improves the overall fitness of the population. Then, the chaotic map is used to optimize the ants' random walk behavior, which improves the ability of the ant population to explore and develop the search space, and helps the algorithm to escape from the local optimal solution. Finally, after the antlion successfully catches the ants, the quasi-opposition learning strategy is utilized to optimize the antlion population position, which improves the algorithm's convergence speed and accuracy of optimization. In the process of Chinese text classification, the traditional feature selection method is first used to preselect the original features, and then CAFSA is used to further select more representative features, thereby lowering the number of features and improving the classification performance of the model. Finally, experiments were conducted on three Chinese data sets. Compare CAFSA with eight different feature selection methods, while each algorithm combines three different classifiers. The results demonstrate that CAFSA performs better overall. It can further select more representative features, which is a more effective feature optimization scheme. The shortcomings of this article are:

1. This experiment only uses three Chinese text data sets, but Chinese data sets from other sources and topics should be evaluated in the future to validate and extend the suggested algorithm.
2. How to make the improved algorithm show better optimization performance is still the next research work.
3. Further apply the algorithm to wider fields such as deep learning.

**Acknowledgements** We thank the editorial board and all reviewers for their professional advises to improve this work.

**Author Contributions** All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by HWC, XZ and DWS. The first draft of the manuscript

was written by HWC and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Funding** This research is funded by the National Natural Science Foundation of China under Grant No. 61772180, the Key R & D plan of Hubei Province (2020BHB004, 2020BAB012) and Natural Science Foundation of Hubei Province No. 2020CFB798.

**Availability of Data and Materials** The datasets used and analysed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** There is no conflict of interest.

**Ethics approval and consent to participate** Not applicable.

**Consent for publication** Not applicable.

**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. Li, Y., Wang, X., Xu, P.: Chinese text classification model based on deep learning. *Future Internet* (2018). <https://doi.org/10.3390/fi10110113>
2. Deng, X., Li, Y., Weng, J., Zhang, J.: Feature selection for text classification: a review. *Multimed. Tools Appl.* **78**(3), 3797–3816 (2019). <https://doi.org/10.1007/s11042-018-6083-5>
3. Huang, C., Zhu, J., Liang, Y., Yang, M., Fung, G.P.C., Luo, J.: An efficient automatic multiple objectives optimization feature selection strategy for internet text classification. *Int. J. Mach. Learn. Cybern.* **10**(5), 1151–1163 (2019). <https://doi.org/10.1007/s13042-018-0793-x>
4. Larabi Marie-Sainte, S., Alalyani, N.: Firefly algorithm based feature selection for arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **32**(3), 320–328 (2020). <https://doi.org/10.1016/j.jksuci.2018.06.004>
5. Yan, L., Zou, F., Guo, R., Gao, L., Zhou, K., Wang, C.: Feature aggregating hashing for image copy detection. *World Wide Web* **19**(2), 217–229 (2016). (**Content based images; Feature aggregation; Image copy detection; Image representations; Large-scale database; Neighborhood structure; Robust feature extractions; Visual word**)
6. Meenachi, L., Ramakrishnan, S.: Metaheuristic search based feature selection methods for classification of cancer. *Pattern Recognit.* **119**, 108079 (2021). (**Ant colony optimization; Feature selection algorithm; Feature selection methods; Fuzzy-rough sets; Medical data**)

- sets;Meta-heuristic search;Neighbourhood;Optimal feature selections;Performance;Search-based;)**
7. Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015). <https://doi.org/10.1016/j.advengsoft.2015.01.010>
  8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
  9. Vergara, J.R., Estévez, P.A.: A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **24**(1), 175–186 (2014). <https://doi.org/10.1007/s00521-013-1368-0>
  10. Zhang, L., Chen, X.: Feature selection methods based on symmetric uncertainty coefficients and independent classification information. *IEEE Access* **9**, 13845–13856 (2021). <https://doi.org/10.1109/ACCESS.2021.3049815>
  11. Lewis, D.D.: Feature selection and feature extraction for text categorization. In: *Proceedings of the Workshop on Speech and Natural Language. HLT '91*, pp. 212–217. Association for Computational Linguistics (1992). <https://doi.org/10.3115/1075527.1075574>
  12. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. *Trans. Neural Netw.* **5**(4), 537–550 (1994). <https://doi.org/10.1109/72.298224>
  13. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of maxdependency, maxrelevance, and minredundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8), 1226–1238 (2005). <https://doi.org/10.1109/TPAMI.2005.159>
  14. Qiu, P., Niu, Z.: Tcic\_fs: total correlation information coefficient based feature selection method for high dimensional data. *Knowl. Based Syst.* **231**, 107418 (2021). <https://doi.org/10.1016/j.knsys.2021.107418>
  15. Moslehi, F., Haeri, A.: An evolutionary computation based approach for feature selection. *J. Ambient Intell. Humaniz. Comput.* **11**(9), 3757–3769 (2020). <https://doi.org/10.1007/s12652-019-01570-1>
  16. Dhar, A., Mukherjee, H., Dash, N.S., Roy, K.: Text categorization: past and present. *Artif. Intell. Rev.* **54**(4), 3007–3054 (2021). <https://doi.org/10.1007/s10462-020-09919-1>
  17. Lu, Y., Liang, M., Ye, Z., Cao, L.: Improved particle swarm optimization algorithm and its application in text feature selection. *Appl. Soft Comput.* **35**, 629–636 (2015). <https://doi.org/10.1016/j.asoc.2015.07.005>
  18. BinSaeedan, W., Alramlawi, S.: Cs-bps: hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis. *Knowl. Based Syst.* **227**, 107224 (2021). <https://doi.org/10.1016/j.knsys.2021.107224>
  19. Asif, M., Nagra, A.A., Ahmad, M.B., Masood, K.: Feature selection empowered by self-inertia weight adaptive particle swarm optimization for text classification. *Appl. Artif. Intell.* (2021). <https://doi.org/10.1080/08839514.2021.2004345>
  20. Ansari, G.J., Shah, J.H., Farias, M.C.Q., Sharif, M., Qadeer, N., Khan, H.U.: An optimized feature selection technique in diversified natural scene text for classification using genetic algorithm. *IEEE Access* **9**, 54923–54937 (2021). <https://doi.org/10.1109/ACCESS.2021.3071169>
  21. Kuancheng, L., Kaiyuan, Z., Huang, Y., Hung, J.C., Yen, N.: Feature selection based on an improved cat swarm optimization algorithm for big data classification. *J. Supercomput.* **72**(8), 3210–3221 (2016). <https://doi.org/10.1007/s11227-016-1631-0>
  22. Thiyagarajan, D., Shanthi, N.: A modified multi objective heuristic for effective feature selection in text classification. *Clust. Comput.* **12**(5), 10625–10635 (2019). <https://doi.org/10.1007/s10586-017-1150-7>
  23. Thirumoorthy, K., Muneeswaran, K.: Optimal feature subset selection using hybrid binary jaya optimization algorithm for text classification. *Sādhanā* **45**(1), 201 (2020). <https://doi.org/10.1007/s12046-020-01443-w>
  24. Janani, R., Vijayarani, S.: Text classification using k-nearest neighbor algorithm and firefly algorithm for text feature selection. In: Sengodan, T., Murugappan, M., Misra, S. (eds.) *Advances in Electrical and Computer Technologies*, pp. 527–539. Springer, Singapore (2020)
  25. Asgarneshad, R., Monadjemi, S.A., Soltanaghaei, M.: An application of mogw optimization for feature selection in text classification. *J. Supercomput.* **77**(6), 5806–5839 (2021). <https://doi.org/10.1007/s11227-020-03490-w>
  26. Ahmad, S.R., Bakar, A.A., Yaakub, M.R.: Ant colony optimization for text feature selection in sentiment analysis. *Intell. Data Anal.* **23**(1), 133–158 (2019). <https://doi.org/10.3233/IDA-173740>
  27. Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A.A., Faris, H.: Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification. *Neural Comput. Appl.* **32**(16), 12201–12220 (2020). <https://doi.org/10.1007/s00521-019-04368-6>
  28. Kyaw, K.S., Limsiraratana, S.: Traditional and swarm intelligent based text feature selection for document classification. In: *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 226–231 (2019). <https://doi.org/10.1109/ISCIT.2019.8905200>
  29. Wang, M., Wu, C., Wang, L., Xiang, D., Huang, X.: A feature selection approach for hyperspectral image based on modified ant lion optimizer. *Knowl. Based Syst.* **168**, 39–48 (2019). <https://doi.org/10.1016/j.knsys.2018.12.031>
  30. Li, M., Qin, W., Zhang, J., Wang, J., Zheng, Q., Liu, Y.: Stable ant-antlion optimiser for feature selection on high-dimensional data. *Electron. Lett.* **57**(3), 106–108 (2021). <https://doi.org/10.1049/el12.12083>
  31. Zawbaa, H.M., Emary, E., Grosan, C., Snasel, V.: Large-dimensionality small-instance set feature selection: a hybrid bio-inspired heuristic approach. *Swarm Evol. Comput.* **42**, 29–42 (2018). <https://doi.org/10.1016/j.swevo.2018.02.021>
  32. Kilic, H., Yuzgec, U., Karakuzu, C.: A novel improved antlion optimizer algorithm and its comparative performance. *Neural Comput. Appl.* **32**(8), 3803–3824 (2020). <https://doi.org/10.1007/s00521-018-3871-9>
  33. Saha, S., Mukherjee, V.: A novel quasi-oppositional chaotic antlion optimizer for global optimization. *Appl. Intell.* **48**(9), 2628–2660 (2018). <https://doi.org/10.1007/s10489-017-1097-7>
  34. Yao, P., Wang, H.: Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle. *Soft Comput.* **21**(18), 5475–5488 (2017). <https://doi.org/10.1007/s00500-016-2138-6>
  35. Dinkar, S., Deep, K.: Opposition-based antlion optimizer using Cauchy distribution and its application to data clustering problem. *Neural Comput. Appl.* **32**(11), 6967–6995 (2020). <https://doi.org/10.1007/s00521-019-04174-0>
  36. Zawbaa, H.M., Emary, E., Grosan, C.: Feature selection via chaotic antlion optimization. *PLoS One* **11**(3), 0150652 (2016). <https://doi.org/10.1371/journal.pone.0150652>
  37. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13**(5), 533–549 (1986). [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)
  38. Tharwat, A., Hassanien, A.E.: Chaotic antlion algorithm for parameter optimization of support vector machine. *Appl. Intell.* **48**(3), 670–686 (2018). <https://doi.org/10.1007/s10489-017-0994-0>
  39. Demidova, L.A., Gorchakov, A.V.: A study of chaotic maps producing symmetric distributions in the fish school search optimization algorithm with exponential step decay. *Symmetry* **12**(5), 784 (2020). <https://doi.org/10.3390/sym12050784>
  40. Yang, D., Liu, Z., Yi, P.: Computational efficiency of accelerated particle swarm optimization combined with different chaotic maps for global optimization. *Neural Comput. Appl.* **28**(S1), 1245–1264 (2017). <https://doi.org/10.1007/s00521-016-2433-2>



41. Zhenxing, Z., Rennong, Y., Huanyu, L., Yuhuan, F., Zhenyu, H., Ying, Z.: Antlion optimizer algorithm based on chaos search and its application. *J. Syst. Eng. Electron.* **30**(2), 352–365 (2019). <https://doi.org/10.21629/JSEE.2019.02.14>
42. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1, pp. 695–701 (2005). <https://doi.org/10.1109/CIMCA.2005.1631345>
43. Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **27**(4), 1053–1073 (2016). <https://doi.org/10.1007/s00521-015-1920-1>
44. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **96**, 120–133 (2016). <https://doi.org/10.1016/j.knosys.2015.12.022>
45. Mahanta, G.B., Rout, A., Deepak, B.B.V.L., Biswal, B.B.: An improved multi-objective antlion optimization algorithm for the optimal design of the robotic gripper. *J. Exp. Theor. Artif. Intell.* **32**(2), 309–338 (2020). <https://doi.org/10.1080/0952813X.2019.1647565>
46. Dhiman, G., Kumar, V.: Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl. Based Syst.* **165**, 169–196 (2019). <https://doi.org/10.1016/j.knosys.2018.11.024>
47. Xue, J., Shen, B.: A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst. Sci. Control Eng.* **8**(1), 22–34 (2020). <https://doi.org/10.1080/21642583.2019.1708830>
48. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. *Future Gener. Comput. Syst.* **97**, 849–872 (2019). <https://doi.org/10.1016/j.future.2019.02.028>

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