

Threshold image segmentation based on improved sparrow search algorithm

Dongmei Wu¹  · Chengzhi Yuan¹

Received: 13 July 2021 / Revised: 2 March 2022 / Accepted: 4 April 2022 /

Published online: 20 April 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022, corrected publication 2022

Abstract

Threshold segmentation based on swarm intelligence optimization algorithm is a research hotspot in image processing, because of its good segmentation effect and easy implementation. This paper proposes an image threshold segmentation method based on an improved sparrow search algorithm and 2-D maximum entropy method. In the proposed algorithm, the nonlinear inertia weight is introduced into the entrants' update formula to improve the local exploration ability of the algorithm, and Levy flight is introduced into the vigilant sparrows' update formula to prevent the algorithm from falling into the local optimal solution in the later stage of iteration. In addition, improved sparrow search algorithm is tested on fifteen benchmark functions. The results represent the merit of the proposed algorithm with respect to other algorithms. Finally, the proposed algorithm is applied to entropy based image segmentation. Experiment results on classical images and medical images show that the proposed method improves the segmentation effect in terms of peak signal-to-noise ratio and feature similarity.

Keywords Swarm optimization algorithm · Threshold image segmentation · 2-D histogram · Maximum entropy · Nonlinear weight · Levy flight

1 Introduction

Threshold image segmentation is a widely used segmentation technology that performs image segmentation based on the information contained in the image gray-level histogram. The gray value difference between the target area and the background area in the image is compared, an appropriate threshold is given to judge the gray value of each pixel in the image belongs to the target region or the background region, so as to segment the image. Nowadays, there are many threshold image segmentation methods, while maximum entropy method [17, 47], inter class

✉ Dongmei Wu
wudm@njupt.edu.cn

¹ Jiangsu Engineering Lab IOT Intelligent Robots, School of Automation & Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China

variance method(OTSU) [4, 41], cross entropy method [10, 25, 50] and minimum error method [5, 22] are divided according to the selection criteria of threshold. Image segmentation based on entropy is very popular, and entropy can be divided into Shannon entropy [39], tsallis entropy [13], exponential entropy [42, 51] and arimoto entropy [30], etc. According to the dimension of image histogram, it also can be classified into one-dimensional (1-D) histogram, two-dimensional (2-D) histogram [1, 6, 45] and so on.

In the process of image threshold segmentation, the effects of some factors on segmentation results are summarized as follows:

- (1) The separation between peaks. It is assumed that the gray level of the image composed of target and background has unimodal distribution characteristics. The gray levels of adjacent pixels in the target or background are highly correlated, but the gray levels of pixels at the junction of target and background are very different. Then the image gray histogram can be regarded as a mixture of two single peak histograms corresponding to the target and background respectively. At this time, if the number of the two distributions is close, the standardized mean difference is far enough and the mean square deviation is small enough, the histogram is bimodal, the pixel value that can be used as the bottom of the valley is found as the threshold between the two peaks. Therefore, the farther the wave crest is away, the greater the chance of separating the target from the background.
- (2) Noise content in the image. Noise will change the gray value of some pixels in the histogram, resulting in imbalance, such as filling the bottom of the histogram and generating new peaks.
- (3) Relative size of objects and background. When the target is small or large, the peak of the target and the background in the histogram will be quite different, so it is difficult to distinguish the valley bottom, which affects the selection of threshold.
- (4) The uniformity of illumination and reflection. When the illumination or the image reflection is uneven, the brightness of the target and background will change, and shadow areas may appear around the target. These factors will affect the change of image gray value, thus affecting the distribution of histogram, resulting in poor segmentation effect.

Using entropy as texture feature is an important research theme, many scholars have used entropy as the basis of threshold image segmentation. Liu et al. proposed a novel image thresholding based on 2-D Arimoto Entropy (TDAE), called Ellipsoid Region Search Strategy (ERSS) [30]. Li et al. proposed a parameter adaptive hybrid model for image segmentation, the image's local entropy was used to establish the index for measuring the image's gray-level information [26]. Chakraborty et al. proposed an improved version of Fractional Order Darwinian PSO (IFODPSO) for segmenting 3D histogram-based color images at multiple levels of Berkley Segmentation Dataset (BSDS500), Multi-level Massi Entropy (MME) had been chosen as the objective function for finding the threshold values in combination with IFODPSO [7]. Nejad et al. proposed a multi-level thresholding method based on imperialist competition algorithm and Kapur's entropy [40]. Yan et al. proposed a image segmentation method based on whale optimization algorithm (WOA) and Kapur's entropy [57]. Lei et al. defined a new form of square rough entropy to measure the roughness in an image, and proposed a corresponding threshold image segmentation algorithm [23]. Moreover, Lin et al. adopted the Tsallis cross-entropy in threshold image segmentation [29].

Only when the most appropriate threshold is found accurately, can the image segmentation be more accurate. Therefore, how to find the appropriate threshold quickly and accurately becomes the key of threshold image segmentation research.

Optimization algorithm based on the swarm intelligence is a bionic algorithm which simulates the behaviors of insects, birds, fish and other wild animals foraging and hunting. There are many well-known algorithms such as particle swarm optimization (PSO) [20], artificial bee colony algorithm (ABC) [18], fruit fly optimization algorithm (FOA) [44], grey wolf optimizer (GWO) [49], chicken swarm optimization (CSO) [36] and the WOA [37] and so on.

In recent years, more and more scholars have used intelligence optimization algorithm to solve image segmentation problems and achieved good results. Some algorithms used in image segmentation are listed in Table 1.

Sparrow search algorithm (SSA) is a new swarm intelligence optimization algorithm proposed by Xue and Shen [56], it is inspired by the foraging behavior and vigilance behavior of sparrows. The algorithm has strong optimization ability and fast convergence speed. It has been widely investigated by scholars. Yuan et al. proposed a DMPPT control of photovoltaic microgrid based on improved SSA [58]. Liu et al. proposed a modified SSA with application in 3-D route planning for UAV [32]. Zhang et al. proposed a semi-supervised ensemble classifier with an improved SSA and applied it to pulmonary nodule detection [59]. Wang et al. conducted a research on economic optimization of microgrid cluster based on chaos SSA [53]. Gai et al. proposed a parameter-optimized deep belief network (DBN) based on SSA for gear fault severity detection [11]. Kathirolu et al. presented an efficient cluster-based routing model using Sparrow search algorithm derives a fitness function to perform clustering process [19]. Tuerxun et al. used the SSA to optimize the penalty factor and kernel function parameter of SVM and constructed the SSA-SVM wind turbine fault diagnosis model [52].

However, the SSA still has the disadvantage of easy to fall into the local optimal solution in the late iteration. Some scholars have put forward different solutions. Yuan et al. proposed an improved SSA based on the center of gravity reverse learning mechanism, learning coefficient and mutation operator [58]. Zhang et al. adopted sine cosine algorithm and new labor cooperation structure to increase the global optimal solution and convergence performance of SSA [59]. Wang et al. proposed a chaos SSA, which combines Bernoulli chaotic mapping, dynamic adaptive weighting, Cauchy mutation, and reverse learning [53]. Li et al. combined the SSA with direction learning strategy and metropolis criterion of simulated annealing algorithm [24]. Lv et al. improved the SSA by combining with the flight behavior in bird swarm algorithm [33]. Zhou et al. proposed a modified SSA by introducing crossover and mutation [60]. This algorithm has higher convergence rate than the original SSA. Liang et al.

Table 1 The summary of image segmentation based on intelligent optimization algorithms

Works	Algorithm
Khairuzzaman [21]	Moth flame optimization algorithm
Kalyani [16]	Exchange market algorithm
Mahajan [34]	Marine predators algorithm
Duan [9]	Cuckoo search algorithm
Liang [27]	Chicken swarm optimization algorithm
Huang [12]	Fruitfly optimization algorithm
Jiang [15]	Teaching-learning-based optimization

modified SSA with combining a homogeneous chaotic system, adaptive inertia weight, and improved boundary constraint [28].

In this research, an improved sparrow search algorithm (ISSA) is proposed which combines Levy flight and nonlinear inertia weight. Nonlinear inertia weight would enhance the local exploration ability and Levy flight would help the algorithm escaping from the local optimum. Then a new image threshold segmentation method is proposed based on ISSA. The gray value of each pixel in the graph and the mean of gray in its field are considered, the 2-D entropy is used as the evaluation function, and the entropy optimization is carried out by ISSA to obtain the threshold corresponding to the maximum entropy. In order to verify the feasibility and effectiveness of ISSA, it is tested on fifteen benchmark functions. Results are compared with some well-known algorithms such as SSA, PSO, GWO, ABC, WOA, CSO, and FOA. In the last, state-of-the-art approaches including ISSA, gold eagle optimizer (GEO) [38], rat swarm optimizer (RSO) [8], transient search optimization (TSO) [46], CRSSA [35], MSCA [48], HHO-PCNN [14], 2D-OTSU [31] and 1D-Exponential entropy [43] are applied to image segmentation. Peak signal-to-noise ratio (PSNR) and feature similarity (FSIM) are used as evaluation indexes for comparison. Experimental results demonstrate the effectiveness of the proposed method.

The main contributions of this paper are:

- (1) An improved sparrow search algorithm combining Levy flight and nonlinear inertia weight factor is proposed.
- (2) Compared with PSO, GWO, ABC, WOA, CSO, FOA and SSA, ISSA has better convergence speed, accuracy and robustness.
- (3) The experimental results show that compared with segmentation methods by GEO, RSO, TSO, CRSSA, MSCA and HHO-PCNN, 2D-OTSU and 1D-Exponential entropy, the segmented images obtained by ISSA and 2-D entropy have excellent peak signal-to-noise ratio and feature similarity.

The remainder of this paper is organized as follows. Section 2 introduces the SSA and the improved method. Section 3 analyzes the convergence of the ISSA. Section 4 proposes the threshold image segmentation based on ISSA and 2-D maximum entropy. Section 5 evaluates the performance of the ISSA and compares the segmentation results for different type of images. Finally, the paper is concluded in Section 6.

2 Improved sparrow search algorithm

2.1 Sparrow search algorithm

Sparrow search algorithm is proposed according to the predatory behavior and vigilant behavior of sparrow population. According to the role of each sparrow in the population, it can be divided into discoverer, entrant and vigilant sparrow.

The discoverers have more energy reserves in the whole population. They are responsible for exploring food for the whole population and guiding the entrants to search for foods. The fitness of the discoverers is better than entrants. In the process of searching for food, if the discoverer detects the danger, it will call to remind other members of the population and move to a new safe area to continue search for foods. In the algorithm, if the fitness of entrant is

better than discoverer, the entrant will become discoverer to search for food. The location update formula of the discoverers is described as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j} \cdot \exp\left(-\frac{i}{\alpha \cdot iter_{max}}\right), & \text{if } R_2 < ST \\ X_{i,j} + Q \cdot L, & \text{if } R_2 \geq ST \end{cases} \tag{1}$$

Where t represents the current iteration, $j = 1, 2, 3, \dots, d$, $iter_{max}$ represents the maximum number of iterations, $X_{i,j}$ represents the position information of the i -th sparrow in the j -th dimension, $\alpha \in [0,1]$ represents a random number, $R_2 \in [0,1]$ represents the warning value, $ST \in [0.5,1]$ represents the safety value, Q represents a random number following normal distribution, L represents a $1 \times d$ matrix, and each element in the matrix is 1.

When the warning value is less than the security value, the discoverers can explore in a large range, and the range gradually decreases with the increase of iterations. When the warning value is bigger than the safety value, the discoverers detect the danger, and then the discoverers randomly go to other safe areas to search for food.

Entrants are the sparrows with the low energy in the population. The sparrows with lower energy are more difficult to find food in their area, so they have to fly to other areas or follow the discoverers to search for foods. Discoverers and entrants will replace each other with the amount of energy stored. The specific location update formula of the entrant is computed as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst} - X_{i,j}^t}{i^2}\right), & \text{if } i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j} - X_p^{t+1}| \cdot A^+ \cdot L, & \text{otherwise} \end{cases} \tag{2}$$

Where X_p is the best position of the current population, X_{worst} is the worst position of the current population, A is the $1 \times d$ matrix, each element in the matrix is randomly 1 or -1 , and $A^+ = A^T(AA^T)^{-1}$. When i is greater than $n/2$, it means that it is very difficult for the sparrows with low energy to find foods, so they need to fly to other places to find foods, otherwise they will fly to the vicinity of the discoverers to get more energy.

In the sparrow population, some sparrows are called vigilant sparrows, which usually account for 10% - 20% of the total population. When they realize the danger, the sparrows near the edge of the group will quickly move towards the safest area to avoid the danger. While the sparrows at the optimal position of the population, it will randomly walk near other sparrows to avoid the danger. The specific vigilant sparrow's position update formula is calculated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t|, & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \epsilon}\right), & \text{if } f_i = f_g \end{cases} \tag{3}$$

Where X_{best} is the optimal position of the current population, β is the step size, which is a normal distribution random number with mean value of 0 and variance of 1, $K \in [-1,1]$, it represents a random number, f_i represents the fitness of the current sparrow, f_g represents the fitness of the current best sparrow, f_w represents the fitness of the current worst sparrow, and ϵ is a small constant to prevent the denominator from being 0.

If the fitness of the current sparrow is greater than the best sparrow, which means that the sparrow is at the edge of the population and is vulnerable to attack, so it needs to move to the

most secure central area. If the fitness of the current sparrow is equal to the best sparrow, which means that the sparrow is in the most secure area. At this time, it needs to move close to the nearby sparrows to avoid danger.

The algorithm flowchart is shown in Fig. 1.

Step 1 Initialize population and algorithm parameters.

Step 2 Calculate the fitness values and sort them from small to large, then divided the population into discoverers and entrants according to the proportion.

Step 3 Update the discoverers' position via Eq.(1).

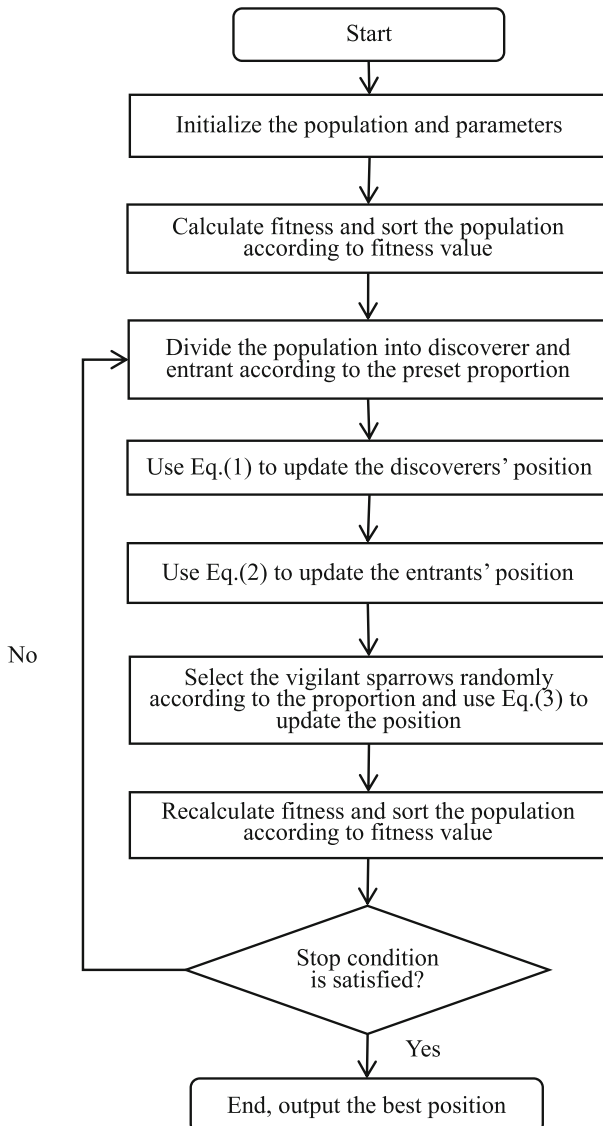


Fig. 1 Flowchart of the SSA

- Step 4 Update the entrants' position use via Eq.(2).
- Step 5 Select some sparrows randomly from the population according to the proportion of vigilant sparrows, and use Eq.(3) to update the position.
- Step 6 Recalculate the new position's fitness. If the fitness is better than that of the previous iteration, update its position, otherwise give up.
- Step 7 Judge whether the stop condition is satisfied. If it is satisfied, exit and output the best position. Otherwise, return to Step 2.

2.2 Improved methods

According to the position updating formula Eq. (2), when the entrants move towards the optimal region, the direction is only 1 or - 1, which makes it easy to fall into the local optimal solution in the late iteration, and the population diversity is poor. Therefore, a new position updating formula is obtained by Eq. (4) with a nonlinear inertia weight of cosine function.

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst} - X_{i,j}^t}{i^2}\right), & \text{if } i > n/2 \\ X_p^{t+1} + w \cdot |X_{i,j} - X_p^{t+1}| \cdot A^+ \cdot L, & \text{otherwise} \end{cases} \tag{4}$$

$$w = w_{max} - (w_{max} - w_{min}) \cos\left(\frac{\pi \cdot T}{G}\right) \tag{5}$$

Where w is the inertia weight, w_{max} and w_{min} are the maximum and minimum values of the weight, T is the number of current iterations, G is the maximum number of iterations. By introducing the nonlinear inertia weight of the cosine function, it can make the entrants explore more carefully and accurately near the current optimal position to make the local exploration ability stronger.

Levy flight was proposed by French mathematician Paul Levy in the 1930s. It is a probability distribution, a random step size that obeys Levy distribution. It is also a walking mode that can transform between a large number of short-distance search and occasional long-distance search. Mantegna algorithm is usually used to simulate, and the specific formula is as follows:

$$u \sim N(0, \delta_u^2) \tag{6}$$

$$v \sim N(0, \delta_v^2) \tag{7}$$

$$\delta_u = \left\{ \frac{\Gamma(1 + \beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta \times 2^{\frac{(\beta-1)}{2}}}\right\}^{\frac{1}{\beta}} \tag{8}$$

$$\delta_v = 1 \tag{9}$$

$$Levy = \frac{\delta_u \times u}{|v|^{\frac{1}{\beta}}} \tag{10}$$

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + Levy \cdot |X_{i,j}^t - X_{best}^t|, & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right), & \text{if } f_i = f_g \end{cases} \tag{11}$$

Where u and v are random numbers of normal distribution, $Levy$ is step size random number of Levy distribution, β is a constant, usually 1.5.

According to the vigilant sparrow’s position update formula, if the current sparrow’s fitness is not optimal, it will move towards the optimal position. The step size control parameter β is a random number of normal distribution with mean value of 0 and variance of 1, which is easy to make the algorithm fall into the local optimal solution in the late iteration. Therefore, the Levy flight random number is introduced to replace β , it will make the algorithm easy to jump out of the local optimal solution, improve the local exploration ability, population diversity, and the optimization performance of the algorithm.

3 Convergence analysis

3.1 Convergence criteria

The ISSA belongs to random search algorithm. Therefore, this paper determines the convergence behavior of ISSA through the convergence criterion of random algorithm [54, 55]. For the optimization problem, there are the random optimization algorithm D , the result of the k -th iteration X_k , and the result of the next iteration $X_{k+1} = D(X_k, \zeta)$, ζ is the searched solution in the iteration of the algorithm D .

The infimum of the search is defined as follows:

$$\varphi = \inf\{t : V_{[x \in A]}[f(x) < t] > 0\} \tag{12}$$

Where $V(x)$ is the Lebesgue measure on the set x , A is the feasible solution space, $f(x)$ is the fitness function.

The optimal solution region can be defined as follows:

$$R_{\varepsilon M} = \begin{cases} \{x \in A | f(x) < \varphi + \varepsilon\}, & \varphi \text{ is limited} \\ \{x \in A | f(x) < -C\}, & \varphi = -\infty \end{cases} \tag{13}$$

Where $\varepsilon > 0$, C is a sufficiently large positive number. If algorithm D finds a point in the optimal solution region, then the algorithm finds an acceptable global optimal solution.

Condition 1: $f(D(x, \zeta)) \leq f(x)$, if $\zeta \in A$, then $f(D(x, \zeta)) \leq f(\zeta)$.

Condition 2: for $B \in A$, s.t. $V(B) > 0$, there is $\prod_{k=0}^{\infty} (1 - u_k(B)) = 0$, where $u_k(B)$ is the probability measure of the k -th iterative search solution of algorithm D on set B .

Let f be measurable, measurable space A be a subset of upper measurable, and algorithm D satisfy condition 1 and condition 2, $\{X_k\}_{k=0}^{\infty}$ is the sequence generated by algorithm D . Then probability measure $\lim_{k \rightarrow \infty} P(x_k \in R_{\epsilon, M}) = 1$, $R_{\epsilon, M}$ is the optimal solution region, and D is the global convergence.

3.2 Convergence analysis of ISSA

The position states of all sparrows in the population constitute the sparrow state, which can be recorded as $s = (x_1, x_2, \dots, x_N)$, where x_i is the position state of the i -th sparrow, N is the number of sparrow population, and all possible states in the sparrow population constitute a state space, which can be recorded as $S = \{s = (x_1, x_2, \dots, x_N) \mid x_i \in A, 1 \leq i \leq N\}$, and A is the feasible solution state space.

In the ISSA, the transition probability of sparrow position state from x_i to x_j is $P(T(x_i) = x_j)$, while the transition probability of sparrow population state from s_i to s_j can be expressed as follows:

$$P(T(s_i) = s_j) = \prod_{k=1}^N P(T(x_{ik}) = x_{jk}) \tag{14}$$

From the above formula, we can know that the transition probability is obtained by multiplying the transition probability of each sparrow, so if $s(t) \in S$, then $s(t + 1) \in S$. The transition probability $P(T(x(t)) = x(t + 1))$ of each sparrow is only related to the state $x(t)$, the number of sparrows i , the maximum iteration number $iter_{max}$, the optimal sparrow position x_p , the worst sparrow position x_{worsts} , and the random number Q at time t , but not related to time t . It can be concluded that the sparrow population position sequence $s(t)$ generated by ISSA has Markov property and homogeneity, and the sparrow population state space s is a finite space. So ISSA's sparrow population sequence $\{s(t), t \geq 0\}$ constitutes a finite homogeneous Markov chain.

In the ISSA, after each iteration, if the fitness value of a new position is better than the current best position, the new position will replace it and become the current best position. The best position of the algorithm is constantly changing and getting better in the iterative process. Therefore, the ISSA satisfies condition 1 of the convergence criterion.

If the optimal solution is x^* , then the optimal state set of sparrow population is $G = \{s \mid f(x) = f(x^*), s \in S\}$, and $G \in S$. for any optimal state set s_i and non optimal state set s_j , the transition probability from s_i to s_j is as follows:

$$P(T(s_i) = s_j) = \prod_{k=1}^N P(T(x_{ik}) = x_{jk}) \tag{15}$$

In the optimal state set G , if at least one state reaches the optimal x^* , then there is a transition probability $P(T(x^*) = x_{jk}) = 0$, then $P(T(s_i) = s_j) = 0$, then we can conclude that the optimal state set G of sparrow population is a closed set on the state space S .

Suppose that there is a nonempty closed set M in the state space S , and $M \cap G = \emptyset$, then for the state $s_i = (x^*)$ in G and the state $s_j = (x_{jk})$ in M , we can get $f(x^*) < f(x_{jk})$, and the transition probability of state s_j to s_i can be known to be greater than 0, so M is not a closed set, so the hypothesis is not true. That is, there is only one closed set in the sparrow population state space S , which is the optimal state set G . That is, the Markov chain of the sparrow population state is irreducible.

Moreover, if there is a nonempty closed set E and there is no other nonempty closed set O to make $E \cap O = \emptyset$, So when $j \in E$, $\lim_{k \rightarrow \infty} P(x_k = j) = \pi_j$, when $j \notin E$, $\lim_{k \rightarrow \infty} p(x_k = j) = 0$, it can be concluded that when the iteration number of the ISSA tends to infinity, the sparrow population state must enter the optimal state set G , the probability that the ISSA can not find the global optimal solution is 0, so the condition 2 of convergence criterion is satisfied, it can be concluded that the ISSA is a global convergence algorithm.

4 Threshold image segmentation

Maximum entropy method was proposed by Kapur in 1985. The threshold selection standard of maximum entropy method is to maximize the total entropy of the segmented target and background (the largest amount of information).

Compared with the 2-D segmentation method, the 1-D segmentation method only considers the gray information of each pixel in the image, so the segmentation speed is faster. In 2-D methods, not only the gray information of each pixel, but also the gray information of its neighborhood is considered. The threshold t and region threshold s are selected to segment the image into four parts. The target and background (A and B in Fig. 2) are mainly concentrated in the two regions near the diagonal, and the other two regions are mainly the image edge and noise interference, we should makes full use of the gray information of the target and background areas, and more efficient and accurate to find the best threshold (t, s) . Although the processing time of this 2-D method is long, the segmentation quality is better and the anti noise ability is superior.

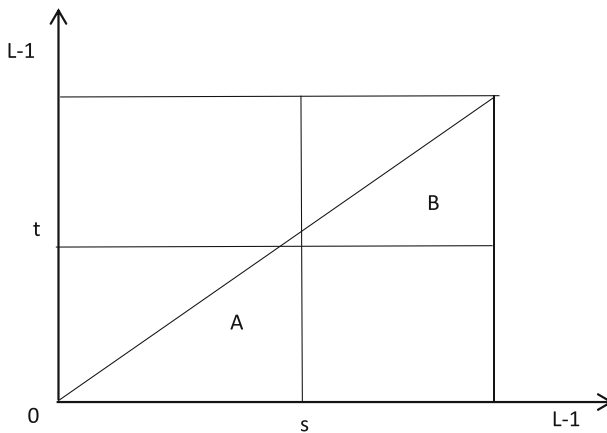


Fig. 2 2-D histogram

Here we take a pixel and its surrounding 3×3 area as a calculation area to calculate the regional gray mean value of a pixel:

$$P_{ij} = \frac{n_{ij}}{M \times N} \tag{16}$$

$$w_0(t, s) = \sum_{i=0}^{t-1} \sum_{j=0}^{s-1} P_{ij} \tag{17}$$

$$w_b(t, s) = \sum_{i=t}^{L-1} \sum_{j=s}^{L-1} P_{ij} \tag{18}$$

Where P_{ij} is the probability when the current pixel gray level is i and its field gray level average is j , n_{ij} is the number of pixels whose gray level value is i and field gray level value is j , $M \times N$ is the pixel size of the image, w_0 and w_b are the probability sum of target class and background class respectively, L is the gray level, $L = 256$.

The image edge and noise interference area information is little and the probability is very small. Thus, we only consider the diagonal target area and background area and set the following formula:

$$w_b(t, s) = 1 - w_0(t, s) \tag{19}$$

For the threshold (t, s) , the entropy expressions of target class and background class are described as follows:

$$H_0(t, s) = - \sum_{i=0}^{t-1} \sum_{j=0}^{s-1} \frac{P_{ij}}{w_0(t, s)} \ln \frac{P_{ij}}{w_0(t, s)} \tag{20}$$

$$H_b(t, s) = - \sum_{i=t}^{L-1} \sum_{j=s}^{L-1} \frac{P_{ij}}{w_b(t, s)} \ln \frac{P_{ij}}{w_b(t, s)} \tag{21}$$

2-D maximum entropy is calculated using Eq. (22).

$$H(t, s) = H_0(t, s) + H_b(t, s) \tag{22}$$

The best threshold is determined as follows:

$$(t^*, s^*) = \operatorname{argmax}(0 \leq t \leq L-1, 0 \leq s \leq L-1) \{H(t, s)\} \tag{23}$$

In the 2-D maximum entropy threshold image segmentation, the ISSA is used to find the best threshold:

$$\operatorname{fun}\{t^*, s^*\} = \operatorname{argmax}(0 \leq t \leq L-1, 0 \leq s \leq L-1) \{H(t, s)\} \tag{24}$$

In the ISSA, the lower the fitness, the better it is, so $-\operatorname{fun}\{t, s\}$ is taken as the fitness function.

The specific segmentation flowchart is shown in Fig. 3, and steps are explained as follows:

Step 1 Read the image to be segmented.

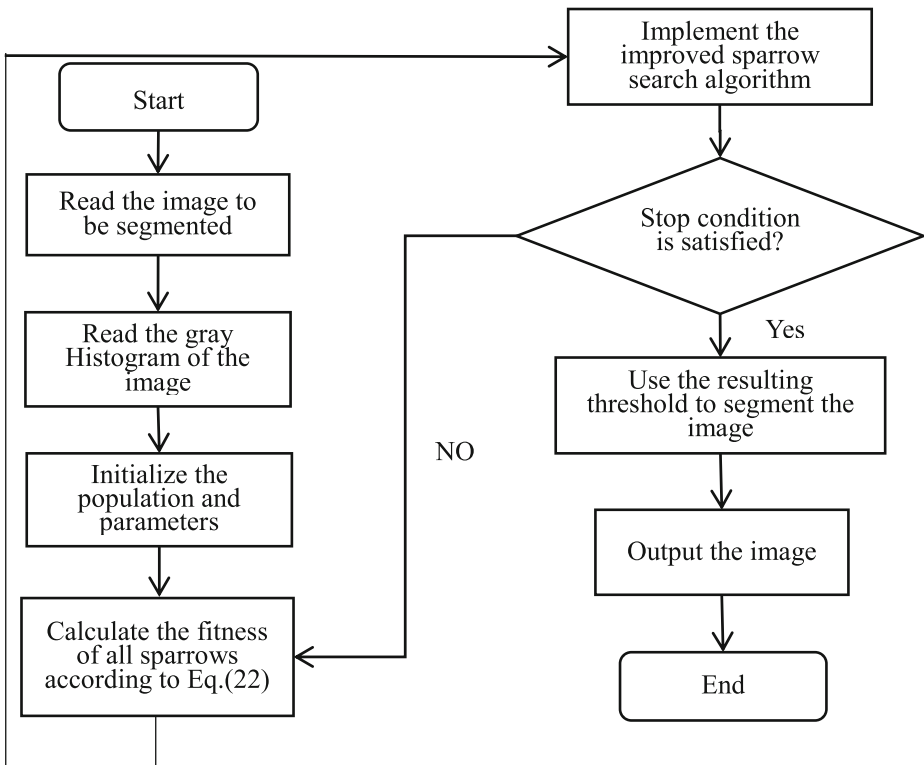


Fig. 3 Image segmentation based on ISSA

Step 2 Read the gray histogram of the image.

Step 3 Initialize the population and parameters.

Step 4 Calculate the fitness of all sparrows according to the Eq. (22).

Step 5 Implement the ISSA.

Table 2 Algorithm parameters

Algorithm	Parameter	Value or range
SSA	Early warning value (ST)	0.6
	Proportion of discoverers (PD)	0.3
	Proportion of vigilant sparrows (PV)	0.2
PSO	Learning factor (C_1, C_2)	2
	Weight (w)	0.8
ABC	Acceleration coefficient upper bound (a)	1
	Abandonment Limit Parameter (L)	$0.6 \times \text{dimension} \times \text{Population size}$
CSO	Proportion of roosters (Pr)	0.15
	Proportion of hens (Ph)	0.7
	Proportion of mother hens (Pm)	0.5
ISSA	Early warning value (ST)	0.6
	Proportion of discoverers (PD)	0.3
	Proportion of vigilant sparrows (PV)	0.2
	Weight (w)	$[0.5, 1.5]$

Table 3 Test functions

Test function	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n X_i^2$	[-100,100]	0
$F_2(x) = \sum_{i=1}^n X_i + \prod_{i=1}^n X_i $	[-10,10]	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i X_j \right)^2$	[-100,100]	0
$F_4(x) = \sum_{i=1}^{n-1} [100(X_{i+1}-X_i^2)^2 + (X_i-1)^2]$	[-30,30]	0
$F_5(x) = \sum_{i=1}^n (X_i + 0.5)^2$	[-100,100]	0
$F_6(x) = \sum_{i=1}^n (X_i^2 - 10\cos(2\pi X_i) + 10)$	[-5.12,5.12]	0
$F_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n X_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi X_i)\right) + 20 + e$	[-32,32]	0
$F_8(x) = \frac{1}{4000} \sum_{i=1}^n X_i^2 - \prod_{i=1}^n \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	[-600,600]	0
$F_9(x) = \frac{\pi}{n} \left\{ 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i-1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n-1)^2 \right\} + \sum_{i=1}^n u(X_i, 10, 100, 4)$	[-50,50]	0
$y_i = 1 + \frac{X_i+1}{4} u(X_i, a, k, m) = \begin{cases} k(X_i-a)^m, X_i > a \\ 0, -a < X_i < a \\ k(-X_i-a)^m, X_i < -a \end{cases}$		
$F_{10}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i-1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n-1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[-50,50]	0
$F_{11}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	[-5,5]	0.0003
$F_{12}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[-5,5]	-1.0316
$F_{13}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	[-2,2]	3
$F_{14}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	[1,3]	-3.86
$F_{15}(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5}{\pi x_1} - 6\right)^2 + 10 \cdot \left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	$lb = [-5,0]$ $ub = [10,15]$	0.39789

Step 6 Judge whether the stop condition is satisfied. If it is satisfied, exit and output the best threshold. Otherwise, return to Step 4.

Step 7 Use the best threshold $\{t^*, s^*\}$ to segment the image and output the segmented image.

Table 4 Test results

Function	Algorithm	Mean	Best	Variance
$F_1(x)$	ISSA	1.2226E-86	0	4.4807E-171
	SSA	1.5136E-81	0	3.1084E-161
	PSO	44.6809	7.5008	504.2381
	GWO	1.5306E-18	4.5467E-20	2.0749E-36
	ABC	222.4236	107.5375	5097.9128
	WOA	1.0761E-49	2.6661E-57	2.5306E-97
	CSO	2.265E-11	2.282E-15	4.9677E-21
	FOA	0.0010302	0.00072754	2.141E-8
$F_2(x)$	ISSA	5.0332E-48	0	7.5295E-94
	SSA	3.2717E-47	0	3.2112E-92
	PSO	2.9053	0.81536	0.73694
	GWO	2.0277E-11	6.3843E-12	1.1796E-22
	ABC	64.3686	23.4021	437.8437
	WOA	1.8813E-32	1.245E-36	4.6361E-63
	CSO	2.4061E-11	5.8686E-13	2.3071E-21
	FOA	1.69	1.4194	0.01227
$F_3(x)$	ISSA	3.577E-61	0	1.8531E-120
	SSA	8.8859E-56	0	2.3671E-109
	PSO	3756.6214	1305.0142	3,070,620.8807
	GWO	0.0030164	5.3608E-5	7.2254E-5
	ABC	70,342.7496	42,937.5191	161,232,308.1177
	WOA	52,860.2692	45,967.0147	16,225,636.2084
	CSO	3838.9673	146.721	3,083,323.408
	FOA	0.28502	0.18773	0.0028213
$F_4(x)$	ISSA	0.00027408	3.0093E-8	3.5113E-7
	SSA	0.00053083	3.5673E-8	2.2185E-6
	PSO	2251.1682	250.9756	11,340,764.6264
	GWO	26.991	25.9835	0.35944
	ABC	1,669,754.7519	756,222.3756	5.41E+11
	WOA	27.9734	27.0766	0.165509
	CSO	27.9994	27.1388	0.19073
	FOA	28.8964	28.5524	0.017629
$F_5(x)$	ISSA	1.7978E-6	5.3044E-12	1.1683E-11
	SSA	5.6235E-6	7.6199E-10	1.6175E-10
	PSO	45.3186	14.282	345.9587
	GWO	0.58243	0.00013016	0.11614
	ABC	198.8354	133.544	1497.052
	WOA	0.29549	0.06514	0.045438
	CSO	3.2061	2.1171	0.17996
	FOA	7.6668	7.6504	0.00010473
$F_6(x)$	ISSA	0	0	0
	SSA	0	0	0
	PSO	39.7484	2.2821	709.5471
	GWO	5.1237	2.2737E-13	20.278
	ABC	249.8306	227.1439	210.8104
	WOA	3.7896E-15	0	2.0798E-28
	CSO	4.9734E-11	0	5.7483E-20
	FOA	72.7537	57.4909	94.9618
$F_7(x)$	ISSA	8.8818E-16	8.8818E-16	0
	SSA	8.8818E-16	8.8818E-16	0
	PSO	3.8735	1.5212	0.5504
	GWO	2.3434E-10	6.8547E-11	2.792E-20
	ABC	7.3415	5.9637	0.73105
	WOA	4.7962E-15	8.8818E-16	4.657E-30
	CSO	6.4677E-7	1.8957E-8	6.8784E-13
	FOA	0.089619	0.066601	6.4997E-5

Table 4 (continued)

Function	Algorithm	Mean	Best	Variance
$F_8(x)$	ISSA	0	0	0
	SSA	0	0	0
	PSO	1.4116	1.0747	0.74327
	GWO	0.0066637	0	0.00033015
	ABC	2.9671	2.1263	0.18277
	WOA	0.016993	0	0.0044624
	CSO	0.024023	1.3634E-13	0.0098685
	FOA	1.5729E-6	1.026E-6	7.5754E-14
	$F_9(x)$	ISSA	2.407E-7	1.6502E-12
SSA		3.778E-7	7.3558E-11	5.9342E-13
PSO		5.0619	0.41142	11.498
GWO		0.034517	0.0065126	0.00056068
ABC		2,226,731.0306	148,783.2411	2.9E+12
WOA		0.02544	0.0017426	0.00066507
CSO		1.2112	0.1207	4.6412
FOA		1.7192	1.7138	1.8251E-5
$F_{10}(x)$		ISSA	2.5115E-6	2.1726E-9
	SSA	3.1115E-6	8.9332E-10	5.2814E-11
	PSO	12.7892	0.85318	128.7476
	GWO	0.49511	0.044972	0.058927
	ABC	7.13E+6	861,494.8983	2.1E+13
	WOA	0.33761	0.057764	0.037903
	CSO	456.177	1.2452	5,859,909.135
	FOA	2.8666	2.8314	0.0002996
	$F_{11}(x)$	ISSA	0.0003076	0.00030749
SSA		0.0003120	0.00030749	1.1009E-10
PSO		0.0003761	0.00030749	1.5625E-8
GWO		0.014444	0.00035003	8.4609E-5
ABC		0.0011008	0.00097179	4.7719E-9
WOA		0.0005307	0.00030764	4.8799E-8
CSO		0.0007175	0.00032418	8.7768E-8
FOA		0.0003892	0.00031135	3.0392E-8
$F_{12}(x)$		ISSA	-1.0316	-1.0316
	SSA	-1.0316	-1.0316	2.6182E-31
	PSO	-1.0316	-1.0316	1.0935E-6
	GWO	-1.0316	-1.0316	1.7009E-15
	ABC	-1.0316	-1.0316	7.3744E-17
	WOA	-1.0316	-1.0316	1.5004E-17
	CSO	-1.0316	-1.0316	1.4171E-15
	FOA	-0.8853	-0.9496	0.0057484
	$F_{13}(x)$	ISSA	3	3
SSA		3	3	5.9913E-29
PSO		3	3	3.4277E-21
GWO		3	3	1.2815E-9
ABC		3	3	6.9683E-18
WOA		3	3	3.178E-9
CSO		3	3	1.4306E-8
FOA		150.7587	84.0352	32,915.0614
$F_{14}(x)$		ISSA	-3.8628	-3.8628
	SSA	-3.8628	-3.8628	6.2624E-27
	PSO	-3.8609	-3.8628	1.5917E-6
	GWO	-3.8614	-3.8628	5.7635E-6
	ABC	-3.8628	-3.8628	1.9454E-28
	WOA	-3.8594	-3.8628	3.0401E-5
	CSO	-3.8611	-3.8628	5.1263E-5
	FOA	-3.6738	-3.8033	0.043112
	$F_{15}(x)$	ISSA	0.39789	0.39789

Table 4 (continued)

Function	Algorithm	Mean	Best	Variance
	SSA	0.39789	0.39789	2.9391E-31
	PSO	0.39789	0.39789	4.0688E-28
	GWO	0.3985	0.39789	1.078E-5
	ABC	0.39789	0.39789	1.19536E-12
	WOA	0.39789	0.39789	8.4979E-12
	CSO	0.39789	0.39789	1.9755E-11
	FOA	21.5722	4.5802	248.406

5 Tests and analysis

5.1 Benchmark test

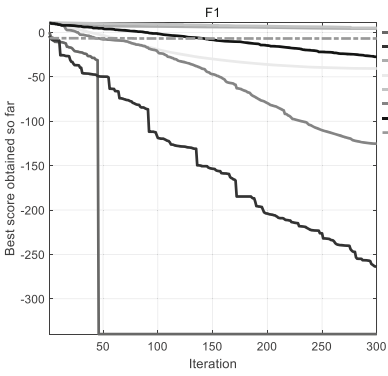
In order to verify the convergence and robustness of the ISSA, fifteen benchmark functions are used to test ISSA and other optimization algorithms such as PSO [29], ABC [20], FOA [18], GWO [44], CSO [49] and WOA [36]. The parameters of these algorithms are shown in Table 2 and the benchmark functions are listed in Table 3. The experiments were performed on the platform of MATLAB R2018a, using a Dell Laptop with an Intel Core i5 processor and 8 GB memory, running the Windows 10 operating system. The number of population is 50 and the iterations is 300, and each algorithm runs 30 times independently.

Table 4 shows the statistical results, it can be seen from Table 4 that for single-mode test functions F_1 - F_5 , the average value, best value and variance of ISSA are smaller than other algorithms, and the global optimal value can be found accurately. For multi-mode test functions F_6 - F_{10} , other algorithms are easy to fall into the local optimal value, but ISSA can accurately find the global optimal solution, which fully reflects excellent optimization ability of ISSA algorithm. For multi-mode test function with fixed dimension F_{11} - F_{15} , most algorithms can accurately find the global optimal solution, but the variance of ISSA is the smallest, which fully reflects excellent robustness of ISSA algorithm.

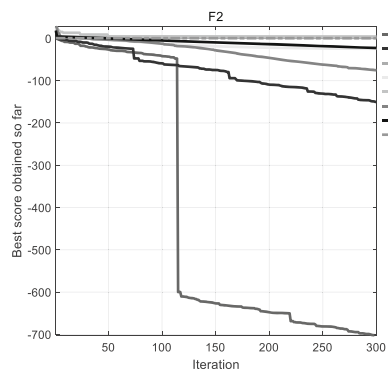
In order to further verify the convergence speed and accuracy of ISSA, the convergence curves of different algorithms are demonstrated in Fig. 4. For the convenience of observation, the ordinate of partial convergence curve is in the form of \log .

It can be seen from Fig. 4 that, for single-mode test function F_1 - F_5 , the convergence of ISSA is obviously better than other algorithms, and the accuracy is higher, especially in the middle and late iteration of ISSA algorithm. For multi-mode test functions F_6 - F_{10} , the convergence performance of ISSA is still better, especially for the F_7 test function. The convergence speed of ISSA is obviously faster than other algorithms in the early iteration. In the F_{10} test function, although the convergence speed of ISSA is not the fastest in the early and middle stages, the convergence accuracy in the later stage is better than other algorithms. For multi-mode test function with fixed dimension F_{11} - F_{15} , although there is little difference in the convergence of each algorithm, and they all converge to the optimal value in the early stage, the convergence speed of ISSA is slightly faster than other algorithms.

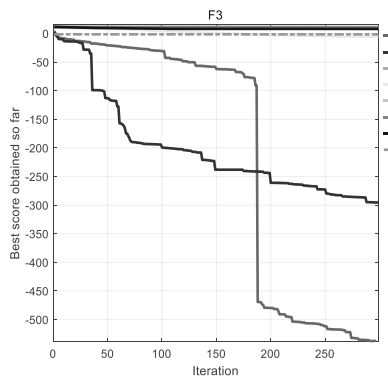
In the complexity test of the algorithm, five single-mode functions with the optimal value of 0 were selected for test. Within 2000 iterations, we recorded the number of iterations and



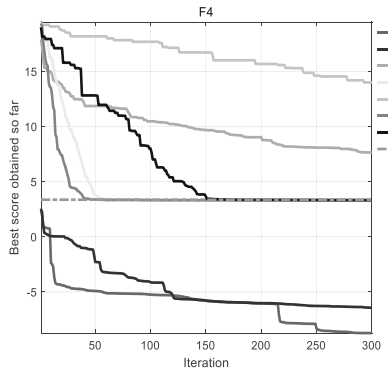
(a) F_1 convergence curve



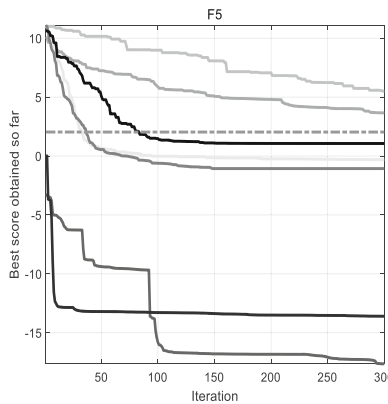
(b) F_2 convergence curve



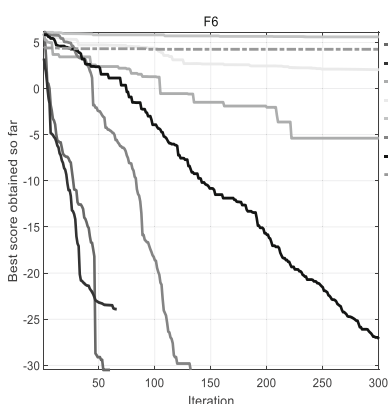
(c) F_3 convergence curve



(d) F_4 convergence curve

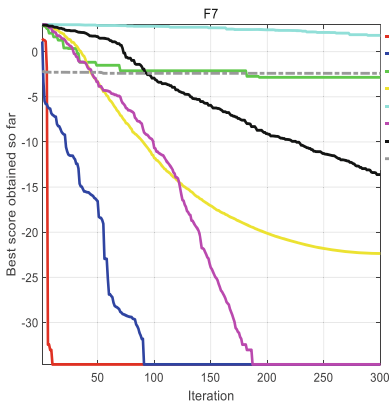


(e) F_5 convergence curve

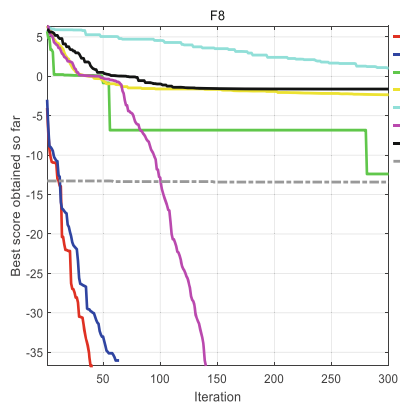


(f) F_6 convergence curve

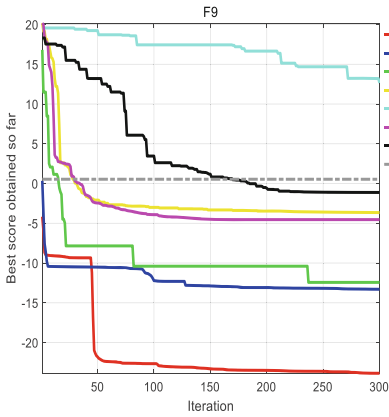
Fig. 4 Convergence curves. **a** F_1 convergence curve **b** F_2 convergence curve **c** F_3 convergence curve **d** F_4 convergence curve **e** F_5 convergence curve **f** F_6 convergence curve **g** F_7 convergence curve **h** F_8 convergence curve **i** F_9 convergence curve **j** F_{10} convergence curve **k** F_{11} convergence curve **l** F_{12} convergence curve **m** F_{13} convergence curve **n** F_{14} convergence curve **o** F_{15} convergence curve



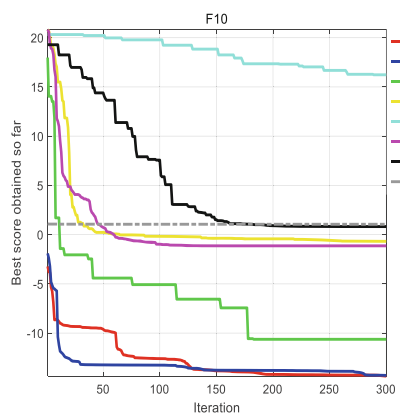
(g) F_7 convergence curve



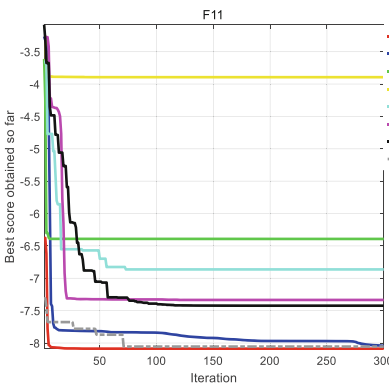
(h) F_8 convergence curve



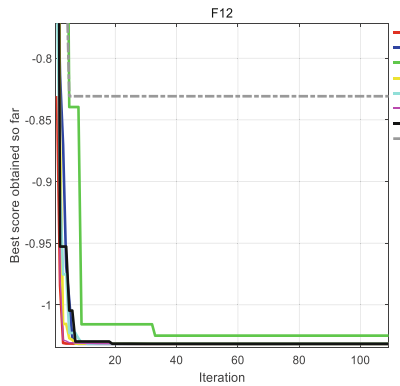
(i) F_9 convergence curve



(j) F_{10} convergence curve



(k) F_{11} convergence curve



(l) F_{12} convergence curve

Fig. 4 (continued)

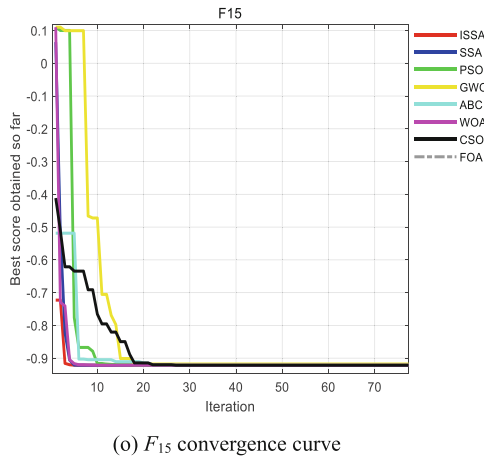
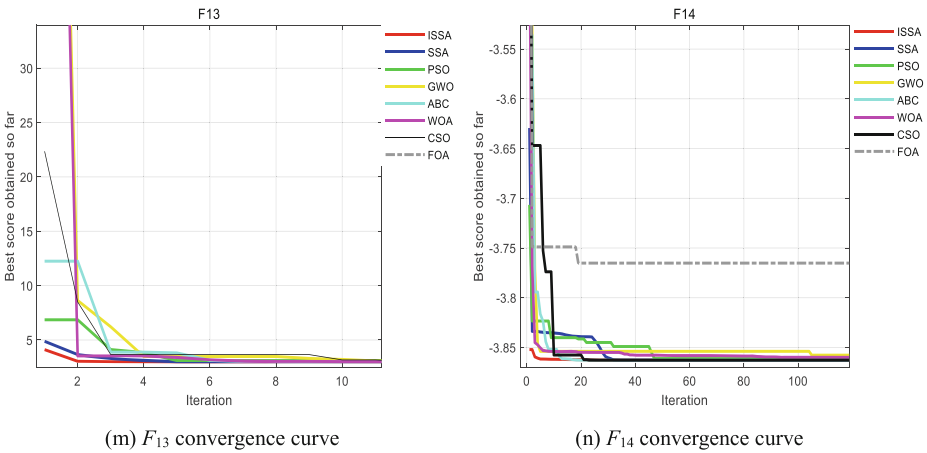


Fig. 4 (continued)

running time when error of the best fitness value is less than 10^{-5} . If the algorithm did not find a solution less than 10^{-5} within 2000 iterations, we recorded the running time of 2000 iterations.

From Table 5, we can see that ISSA can converge below 10^{-5} with the least number of iterations on the five test functions, and in general, ISSA has lower computation complexity, except F_2 . It can be concluded that under the same number of iterations, ISSA will spend a longer running time, but it makes up for this drawback with an excellent optimization ability.

From the above test results, we can draw a conclusion that the convergence speed, convergence accuracy and robustness of ISSA algorithm are superior to other seven algorithms, and it has better optimization ability.

Table 5 Complexity test results

Function	Algorithm	Number of iterations	Running time (s)
$F_1(x)$	ISSA	3	0.037
	SSA	11	0.045
	PSO	>2000	>0.566
	GWO	69	0.059
	ABC	1390	3.925
	WOA	85	0.058
	CSO	155	0.051
	FOA	>2000	0.675
$F_2(x)$	ISSA	22	0.066
	SSA	38	0.083
	PSO	>2000	0.574
	GWO	77	0.071
	ABC	1459	4.139
	WOA	73	0.057
	CSO	145	0.519
	FOA	>2000	>0.672
$F_3(x)$	ISSA	11	0.061
	SSA	42	0.119
	PSO	>2000	>1.673
	GWO	338	0.354
	ABC	>2000	>7.813
	WOA	>2000	>2.015
	CSO	>2000	>2.474
	FOA	>2000	>1.705
$F_4(x)$	ISSA	10	0.052
	SSA	72	0.127
	PSO	>2000	>0.715
	GWO	>2000	>0.969
	ABC	>2000	>5.914
	WOA	>2000	>0.829
	CSO	>2000	>1.549
	FOA	>2000	>0.822
$F_5(x)$	ISSA	23	0.066
	SSA	42	0.087
	PSO	>2000	>0.559
	GWO	>2000	>0.848
	ABC	1358	3.881
	WOA	>2000	>0.669
	CSO	>2000	>1.369
	FOA	>2000	>0.673

Table 6 Parameters of the partial algorithms

Algorithms	Parameters	Value or range
GEO	Attack Propensity (P_a)	[0.5,2]
	Cruise propensity (P_c)	[1,0.5]
CRSSA	Early warning value (ST)	0.6
	Proportion of discoverers (PD)	0.7
	Proportion of vigilant sparrows (PV)	0.2
MSCA	Proportion of main groups (PM)	0.5
HHO-PCNN	Boundary range	[0.001,200]

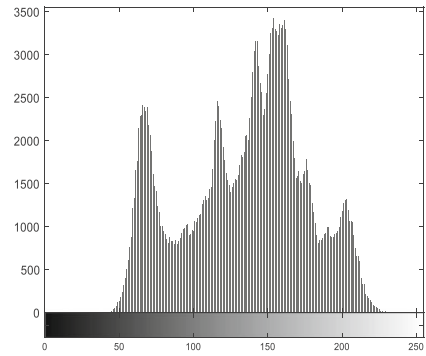
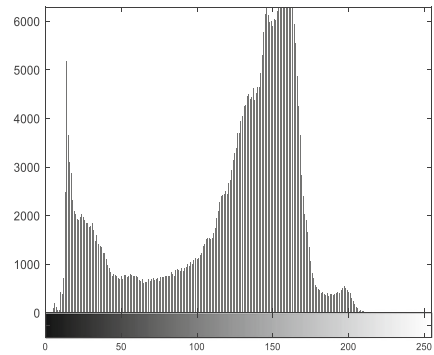
(a) Lena(512×512)(b) Boat(720×576)

Fig. 5 The test images and corresponding histograms. **a** Lena(512×512) **b** Boat(720×576) **c** Zelda(512×512) **d** Aircraft(512×512) **e** Monkey(500×480) **f** Flower(512×480) **g** Butterfly(768×512) **h** Parrot(768×512) **i** Bear(394×600) **j** Gaussian noise(512×512) **k** Salt pepper noise(512×480) **l** Poisson noise(768×512)

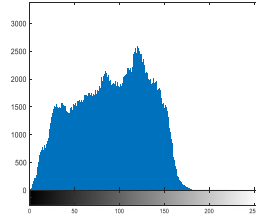
5.2 Application to threshold image segmentation

In this section, the ISSA is applied to threshold image segmentation, the optimal threshold and image segmentation quality are compared with image segmentation results based on GEO [38], RSO [8], TSO [46], CRSSA [35], MSCA [48], HHO-PCNN [14], 2D-OTSU method [31] and 1D-Exponential entropy method [43]. The maximum number of iterations is 30, the population size is 20. The parameters to be set for algorithms are shown in Table 6.

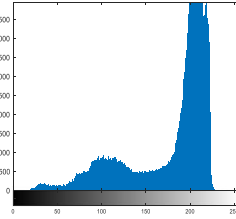
5.2.1 Application to classical images

We selected three gray images, six color images and three images with different noises for experiments. The test images and their resolutions and gray histograms are shown in Fig. 5.

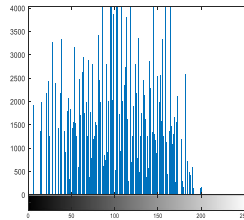
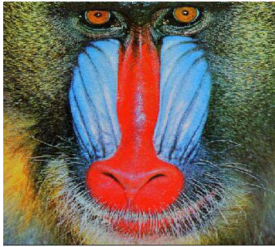
From the results in Table 7, we can see that the proposed method is obviously superior to other methods. In twelve images, the threshold values sought by eleven images have higher



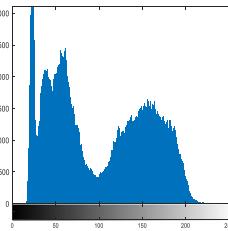
(c) Zelda(512×512)



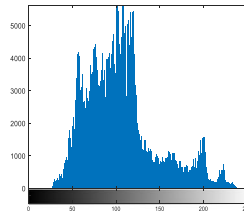
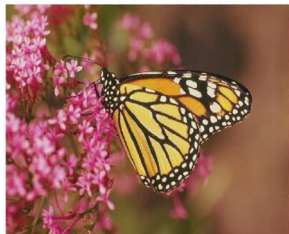
(d) Aircraft(512×512)



(e) Monkey(500×480)

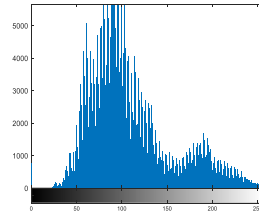


(f) Flower(512×480)

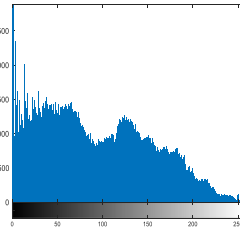


(g) Butterfly(768×512)

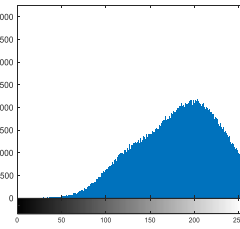
Fig. 5 (continued)



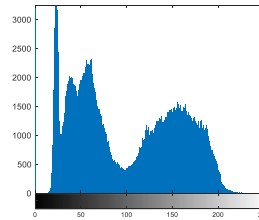
(h) Parrot(768 × 512)



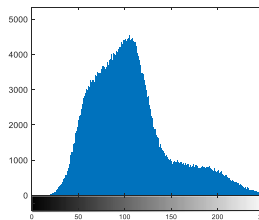
(i) Bear(394 × 600)



(j) Gaussian noise(512 × 512)



(k) Salt pepper noise(512 × 480)



(l) Poisson noise(768 × 512)

Fig. 5 (continued)

Table 7 Comparison of the threshold and 2-D maximum entropy by different algorithm

Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA
Lena	203.8,113.4 13.5245	103.0,1.0 13.4806	108.1,177.5 13.5108	111.7,166.8 13.5196	112.4,154.6 13.5196	183.1,113.4 13.5245
Boat	172.9,113.5 14.3852	102.4,140.3 14.3355	108.1,177.5 14.3666	113.4,161.2 14.3769	112.8,254.5 14.3769	247.9,112.9 14.3852
Zelda	167.6,90.3 13.7856	255.0,69.5 13.7854	233.6,81.5 13.7757	90.0,220.6 13.7856	240.0,89.7 13.7856	89.6,199.9 13.7856
Aircraft	224.9,155.5 14.1608	145.9,103.4 14.1494	161.6,173.4 14.1402	238.0,155.2 14.1608	202.2,155.3 14.1607	234.9,154.5 14.1608
Monkey	92.8,127.3 15.7505	242.4,30.3 15.7405	139.9,100.6 15.7176	183.5,94.5 15.7449	93.7,180.6 15.7610	95.3,189.9 15.7610
Flower	123.2,123.3 13.3982	189.8,156.8 13.3928	108.1,177.5 13.3889	126.8,121.9 13.3955	118.3,247.2 13.3936	120.4,118.5 13.3976
Butterfly	141.2,140.5 14.1197	141.4,96.3 14.0986	134.3,135.9 14.0961	141.2,142.1 14.1179	141.2,245.2 14.0954	139.5,140.2 14.1190
Parrot	141.9,141.8 13.9767	237.9,171.1 13.9426	167.5,149.7 13.9197	142.3,143.3 13.9755	255.0,140.3 13.9451	141.9,142.7 13.9755
Bear	137.2,137.2 15.3526	133.2,244.7 15.3288	108.1,177.5 15.2805	138.0,136.7 15.3523	139.2,135.2 15.3474	135.5,134.8 15.3520
Gaussian	254.5,164.4 17.3528	155.5,123.1 17.2974	163.0,242.1 17.3411	161.7,239.9 17.3413	162.4,236.9 17.3413	255.0,163.0 17.3528
Salt pepper	128.3,128.1 15.1855	191.4,234.4 15.1730	204.8,123.0 15.1736	134.1,124.8 15.1806	205.7,122.5 15.1737	128.6,127.0 15.1851
Poisson	146.5,143.3 16.7802	141.3,87.2 16.7599	162.2,148.5 16.7539	144.6,143.8 16.7792	141.6,145.0 16.7721	145.1,144.2 16.7801

Table 8 Comparison of the PSNR

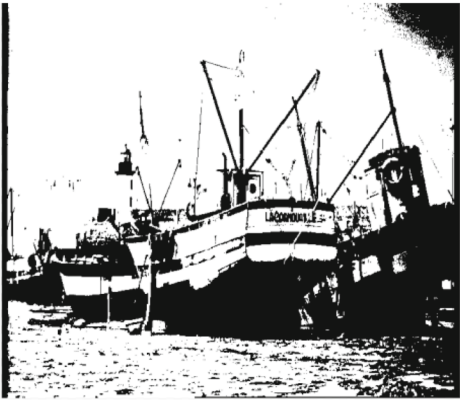
Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA	HHO-PCNN	2D-OTSU	1D-Exponential entropy
Lena	8.6121	8.2058	8.5733	8.5890	8.3434	8.4879	5.1967	5.2056	5.2682
Boat	8.5029	8.2671	8.3332	8.4565	8.4542	8.2940	5.9079	5.9105	5.9871
Zelda	8.8047	8.4057	8.7959	8.5545	8.7099	8.4973	8.2054	8.2061	8.2918
Aircraft	11.3807	11.2832	10.7475	10.8704	10.6266	10.5807	2.8095	2.8283	2.8683
Monkey	8.9211	8.6151	8.8474	8.6235	8.9220	8.9233	7.1941	7.1913	7.2740
Flower	10.1489	9.523	10.1306	10.1436	10.0597	10.1129	6.9952	6.9961	7.0573
Butterfly	8.8844	8.7701	8.8790	8.8517	8.4690	8.8701	7.0546	7.0608	7.1038
Parrot	9.1388	8.6748	8.7554	9.1132	8.3418	9.1028	6.6706	6.6554	6.7189
Bear	10.5302	10.3558	10.5004	10.5261	10.4259	10.5097	7.5177	7.5036	7.5676
Gaussian	8.3671	8.3119	8.1361	8.2558	8.2616	8.1982	2.5837	5.0494	5.1293
Salt pepper	10.3875	8.3756	9.2679	9.5360	10.0583	10.3839	6.6713	6.6719	6.7511
Poisson	8.9017	8.7533	8.5923	8.7897	8.8071	8.8905	7.0177	6.9889	7.0656

Table 9 Comparison of the FSIM

Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA	HHO-PCNN	2D-OTSU	1D-Exponential entropy
Lena	0.6276	0.6120	0.6175	0.6189	0.5953	0.6086	0.5314	0.5495	0.5484
Boat	0.5839	0.5109	0.5156	0.5248	0.5236	0.5128	0.4767	0.4710	0.4421
Zelda	0.5353	0.5225	0.5351	0.5176	0.5080	0.5184	0.5437	0.5447	0.5442
Aircraft	0.6843	0.6361	0.6579	0.6716	0.6580	0.6582	0.5095	0.5308	0.4481
Monkey	0.5526	0.5388	0.5450	0.5531	0.5574	0.5610	0.3341	0.3430	0.3342
Flower	0.6215	0.5699	0.6009	0.6088	0.5918	0.6209	0.5365	0.5477	0.5217
Butterfly	0.6518	0.6101	0.6356	0.6484	0.6475	0.6461	0.4792	0.4784	0.4562
Parrot	0.6686	0.6610	0.6638	0.6711	0.6500	0.6473	0.6185	0.6480	0.6528
Bear	0.4514	0.4129	0.4494	0.4473	0.4278	0.4301	0.3089	0.2850	0.2874
Gaussian	0.5985	0.4978	0.4838	0.4850	0.4849	0.5926	0.3379	0.3130	0.3090
Salt pepper	0.6593	0.5614	0.5603	0.5679	0.5835	0.6587	0.3179	0.3131	0.3199
Poisson	0.6071	0.5595	0.5935	0.5927	0.5911	0.6061	0.4105	0.4423	0.3812



(a) Lena



(b) Boat



(c) Zelda



(d) Aircraft

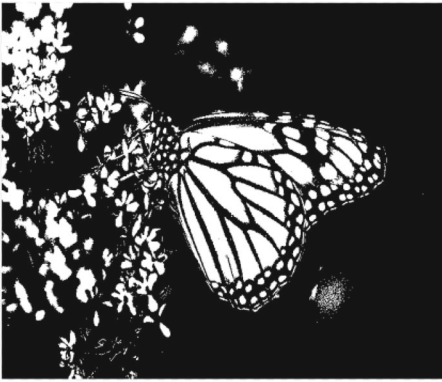


(e) Monkey

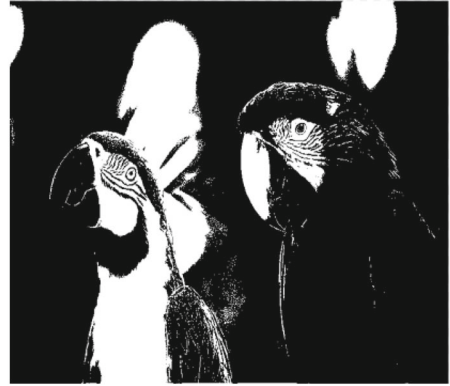


(f) Flower

Fig. 6 Segmented images. **a** Lena **b** Boat **c** Zelda **d** Aircraft **e** Monkey **f** Flower **g** Butterfly **h** Parrot **i** Bear **j** Gaussian noise **k** Salt pepper noise **l** Poisson noise



(g) Butterfly



(h) Parrot



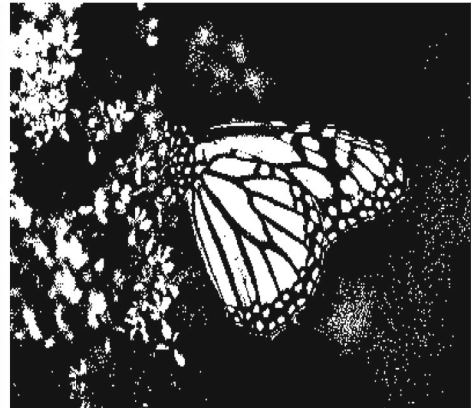
(i) Bear



(j) Gaussian noise

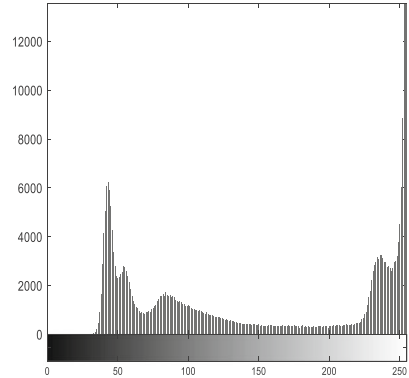
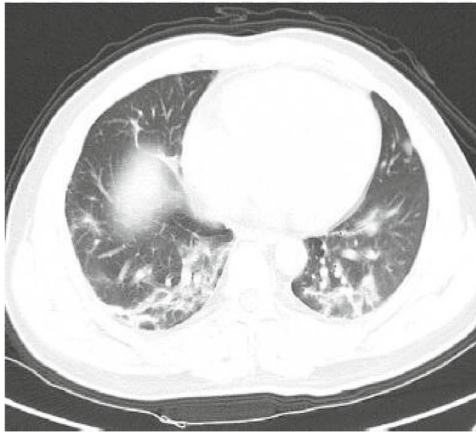


(k) Salt pepper noise

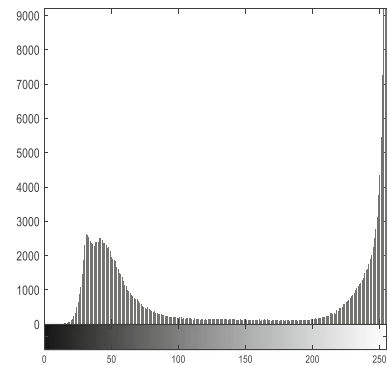


(l) Poisson noise

Fig. 6 (continued)



(a) Lung 1(725 × 551)



(b) Lung 2(570 × 436)

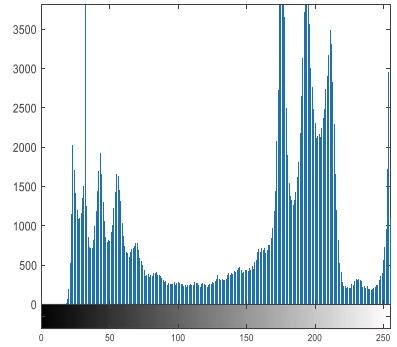
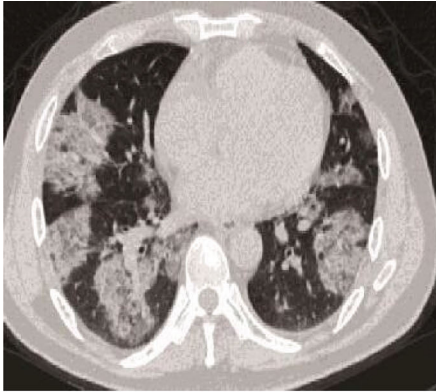
Fig. 7 Lung CT images and corresponding histograms. **a** Lung 1(725 × 551) **b** Lung 2(570 × 436) **c** Lung 3(563 × 437) **d** Lung 4(556 × 398) **e** Lung 5(457 × 285)

entropies, which reflects the good convergence and high robustness of the ISSA-based segmentation method. Although MSCA-based segmentation method has also obtained a good threshold, on the whole, the performance of the ISSA-based segmentation method is more prominent.

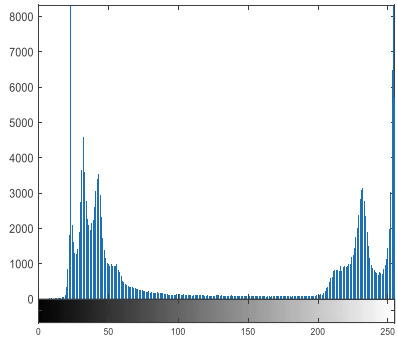
Tables 8 and 9 show the PSNR and FSIM of the nine methods for twelve images. It can be seen that the threshold obtained by ISSA is used for image segmentation, although the PSNR and FSIM of some images are not ideal, the overall performance of ISSA-based segmentation is better. The segmented images are shown in Fig. 6.

5.2.2 Application to medical images

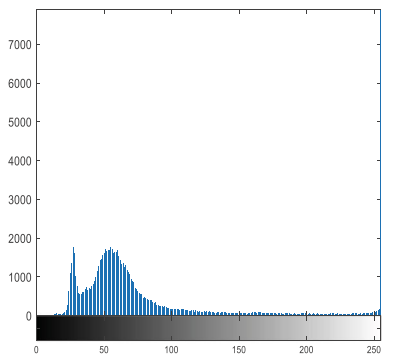
The outbreak of COVID-19 has brought great harm to people all over the world, and countless people have lost their lives. Image segmentation is also widely used in the medical field. For medical images, how to segment accurately is particularly important [3]. For lung CT images,



(c) Lung 3(563 × 437)



(d) Lung 4(556 × 398)



(e) Lung 5(457 × 285)

Fig. 7 (continued)

Table 10 Comparison of the threshold and 2-D maximum entropy

Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA
Lung1	108.1,177.5 11.5342	108.4,54.6 11.5342	108.1,114.2 11.5267	108.2,141.5 11.5341	108.1,247.4 11.5342	108.1,244.8 11.5342
Lung2	108.2,202.5 12.6203	106.2,134.6 12.6202	108.1,177.5 12.6203	112.3,184.8 12.6202	108.1,247.4 12.6203	107.8,228.4 12.6202
Lung3	205.9,165.6 13.9072	165.2,48.1 13.9028	244.2,167.5 13.9068	186.4,165.8 13.9070	164.6,250.4 13.9028	253.2,165.9 13.9065
Lung4	131.1,173.9 10.8967	172.4,96.4 10.8966	108.1,177.5 10.8935	215.1,129.0 10.8967	131.4,254.5 10.8967	131.3,181.2 10.8967
Lung5	22.5,114.5 11.1086	49.3,86.7 11.0683	23.9,101.7 11.0485	50.9,85.8 11.0365	47.6,105.9 11.0686	22.7,146.8 11.1084

Table 11 Comparison of the PSNR

Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA	HHO-PCNN	2D-OTSU	1D-Exponential entropy
Lung1	13.6623	13.3973	13.5342	13.6315	12.8215	11.8053	2.3953	2.3994	2.4437
Lung2	16.0535	16.0065	15.1025	16.0093	15.1347	14.7924	2.3581	2.3636	2.4053
Lung3	11.7959	11.0640	10.2886	10.4384	11.0714	11.0640	3.8973	3.9088	3.9530
Lung4	16.9460	16.4824	16.2028	16.6810	16.8129	16.5475	2.9154	2.9218	2.9604
Lung5	14.1732	13.0654	13.7156	13.1242	13.8251	14.1640	3.5104	3.5112	3.5615

Table 12 Comparison of the FSIM

Images	ISSA	GEO	RSO	TSO	CRSSA	MSCA	HHO-PCNN	2D-OTSU	1D-Exponential entropy
Lung1	0.6994	0.6682	0.6711	0.6808	0.7148	0.6929	0.5628	0.5655	0.5793
Lung2	0.7064	0.7044	0.6968	0.6967	0.7053	0.7021	0.4855	0.4883	0.4682
Lung3	0.5949	0.5351	0.5324	0.4929	0.5332	0.5351	0.4118	0.4305	0.4065
Lung4	0.7309	0.7191	0.7084	0.7244	0.7279	0.7208	0.5291	0.5315	0.5295
Lung5	0.6835	0.6118	0.6423	0.6395	0.6484	0.6798	0.5067	0.5142	0.5155

the primary task of image segmentation is to segment the lung organs and nodules in the lung, which can enable doctors to judge the situation of the lung more accurately and prevent wrong judgment caused by blurred images [2]. Therefore, we used five lung CT images of COVID-19 patients to verify the effectiveness of the proposed method. The CT images, as well as the corresponding resolution and gray histogram are shown in Fig. 7 [2].

Tables 10, 11, 12 show a comparison of the thresholds, optimal entropy, PSNR and FSIM of different methods for five lung images. It is noted from the tables that other algorithms can find the threshold corresponding to the maximum entropy, but the effect of segmentation using these obtained threshold is not very ideal. Since ISSA can not only find the threshold of maximum entropy, but also the segmented images based on the found threshold have larger PSNR and FSIM.

Figure 8 shows the segmented images obtained by ISSA method. It can be seen from Fig. 8 that the lung organs are completely segmented, and the nodules and diseased tissues in the

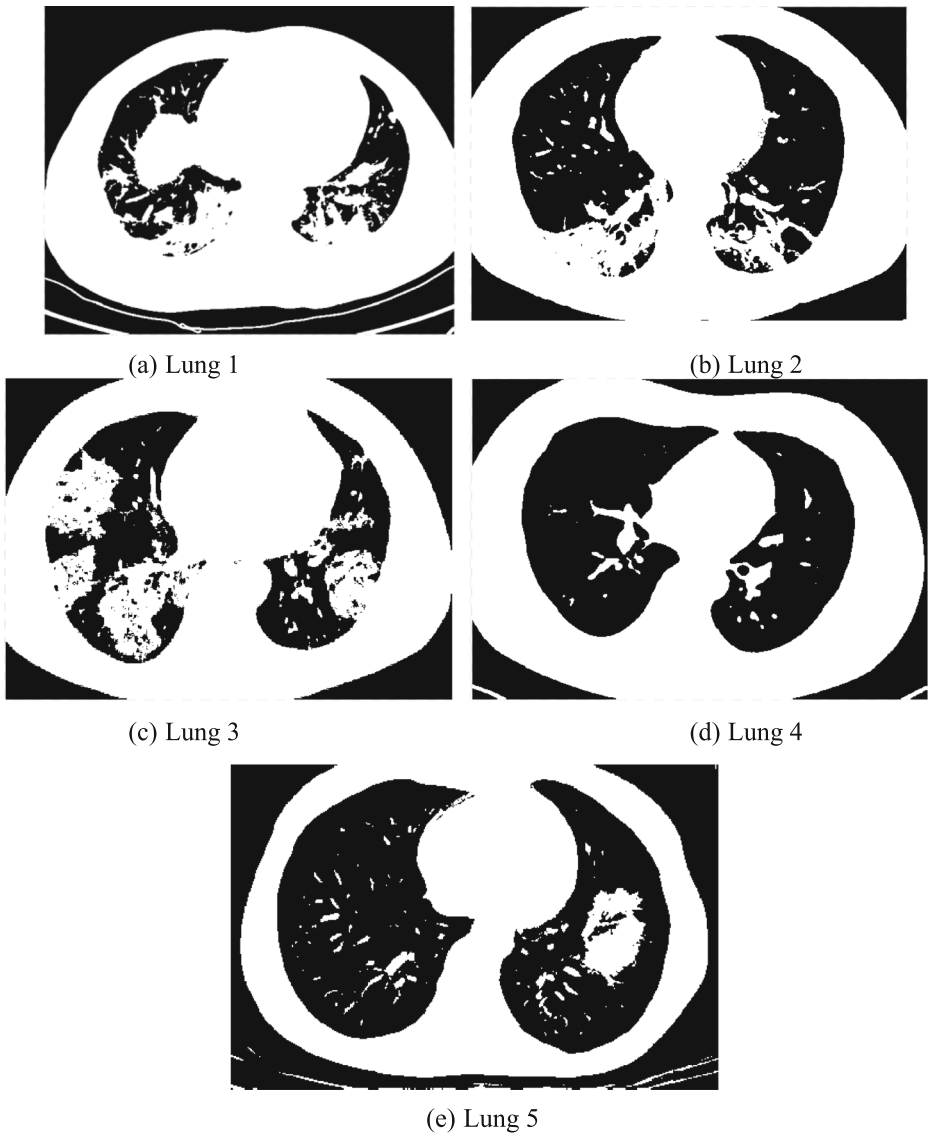


Fig. 8 Lung CT segmentation images. **a** Lung 1 **b** Lung 2 **c** Lung 3 **d** Lung 4 **e** Lung 5

lung are clearly distinguished, which plays a certain role in the diagnosis of COVID-19 patients.

Table 13 Number of times to get the highest PSNR

Methods	3 gray images	6 color images	3 noise images	5 medical images
ISSA	3	5	3	5
MSCA	0	1	0	0

Table 14 Number of times to get the highest FSIM

Methods	3 gray images	6 color images	3 noise images	5 medical images
ISSA	2	4	3	4
TSO	0	1	0	0
CRSSA	0	0	0	1
MSCA	0	1	0	0
2D-OTSU	1	0	0	0

6 Conclusion

This paper proposes an ISSA by combining the nonlinear inertia weight and Levy flight and applies the ISSA to threshold image segmentation based on 2-D maximum entropy. The experimental results deliver several conclusions. First, compared with other algorithms, ISSA has faster convergence speed, more accurate convergence results and stronger robustness. Second, Through the threshold obtained by ISSA optimization for classical images segmentation, the maximum entropy is obtained, and higher PSNR and FSIM are achieved. The quality of segmented images is better. Third, the segmentation results for lung CT images show that in medical application, ISSA can still obtain better performance, including higher PSNR and FSIM. The nodules and diseased tissues on lung organs can be clearly segmented.

Tables 13 and 14 list algorithms that obtained the highest PSNR and FSIM in experiments, and record number of times to obtain the optimal values when segmented different images. The statistical results show the ISSA performs better in image segmentation of different types of images. Since in most tests, PSNR and FSIM obtained using ISSA method are the best.

The experiment results confirm the effectiveness of the proposed method, but there are some drawbacks. The running time of the ISSA increases compared with SSA, and in test function F_7 , ISSA cannot converge to the global optimal 0.

Image segmentation remains one of the main issues in image processing, this study deserves further methodological, algorithmic and numerical investigations. And as a newly proposed swarm intelligence optimization algorithm, the SSA has strong robustness, fast convergence speed, and can be applied to optimization problems in other areas, so it has great research significance. In future work, we will continue to focus on improvement of swarm intelligence optimization algorithm, and combine with deep learning method to solve image processing problems.

Acknowledgments The authors would like to thank the Editor-in-Chief, the Associate Editor, and the reviewers for their insightful comments and suggestions.

Funding This work is supported by National Natural Science Foundation of China under Grant 61873326, Natural Science Foundation of Jiangsu Province under Grant BK20130873, The Scientific Research Foundation for Returned Scholars of Ministry of Education under Grant BJ213010, The Scientific Research Funds of Nanjing University of Posts and Telecommunications under Grant NY220216.

Declarations

Conflict of interest We confirm that this article content has no conflicts of interest to declare.

References

1. Abutaleb A, Eloteifi A (1988) Automatic thresholding of gray-level pictures using 2-D entropy. In: 31st annual technical symposium. International Society for Optics and Photonics. <https://doi.org/10.1117/12.942103>
2. Afshin S, Marjane K, Roohallah A et al (2020) Automated detection and forecasting of COVID-19 using deep learning techniques: a review. arXiv preprint arXiv:2007.10785
3. Afshin S, Marjane K, Mahboobeh J et al (2021) Applications of deep learning techniques for automated multiple sclerosis detection using magnetic resonance imaging: a review. *Comput Biol Med* 136:104697. <https://doi.org/10.1016/j.combiomed.2021.104697>
4. Ashish K, Bhandari A, Ghosh I et al (2020) A local contrast fusion based 3d otsu algorithm for multilevel image segmentation. *IEEE/CAA J Autom Sin* 7(01):203–216. <https://doi.org/10.1109/JAS.2019.1911843>
5. Barron JT (2020) A generalization of Otsu's method and minimum error thresholding. In: Vedaldi A, Bischof H, Brox T, Frahm JM (eds) *Computer Vision – ECCV 2020*. ECCV 2020. Lecture notes in computer science, vol 12350. Springer, Cham. https://doi.org/10.1007/978-3-030-58558-7_27
6. Cao Q, Shi Z, Wang R, Wang P, Yao S (2020) A brightness-preserving two-dimensional histogram equalization method based on two-level segmentation. *Multimed Tools Appl* 79(2):27091–27114. <https://doi.org/10.1007/s11042-020-09265-y>
7. Chakraborty R, Verma G, Namasudra S (2020) Ifodpso-based multi-level image segmentation scheme aided with masi entropy. *J Ambient Intell Humaniz Comput* 1:1–19. <https://doi.org/10.1007/s12652-020-02506-w>
8. Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2021) A novel algorithm for global optimization: rat swarm optimizer. *J Ambient Intell Humaniz Comput* 12(8):8457–8482. <https://doi.org/10.1007/s12652-020-02580-0>
9. Duan L, Yang S, Zhang D (2021) Multilevel thresholding using an improved cuckoo search algorithm for image segmentation. *J Supercomput* 77(7):6734–6753. <https://doi.org/10.1007/s11227-020-03566-7>
10. Esmaili L, Mousavirad SJ, Shahidi Ne Jad A (2021) An efficient method to minimize cross-entropy for selecting multi-level threshold values using an improved human mental search algorithm. *Expert Syst Appl* 182(3):115106. <https://doi.org/10.1016/j.eswa.2021.115106>
11. Gai J, Zhong K, Du X, Yan K, Shen J (2021) Detection of gear fault severity based on parameter-optimized deep belief network using sparrow search algorithm. *Measurement* 185:110079. <https://doi.org/10.1016/j.measurement.2021.110079>
12. Huang C, Li X, Wen Y (2020) An otsu image segmentation based on fruitfly optimization algorithm. *AEJ - Alex Eng J* 60(1):183–188. <https://doi.org/10.1016/j.aej.2020.06.054>
13. Jena B, Naik MK, Panda R, Abraham A (2021) Maximum 3d tsallis entropy based multilevel thresholding of brain mr image using attacking manta ray foraging optimization. *Eng Appl Artif Intell* 103(16):104293. <https://doi.org/10.1016/j.engappai.2021.104293>
14. Jia H, Zhang LF, Sun KJ et al (2019) Harris eagle algorithm optimized automatic image segmentation based on pulse coupled neural network. *Appl Sci Technol* 46(4):16–20+25. <https://doi.org/10.11991/yykj.201904022>
15. Jiang Z, Zou F, Chen D, Kang J (2021) An improved teaching–learning-based optimization for multilevel thresholding image segmentation. *Arab J Sci Eng* 46:8371–8396. <https://doi.org/10.1007/s13369-021-05483-0>
16. Kalyani R, Sathya PD, Sakthivel VP (2021) Multilevel thresholding for image segmentation with exchange market algorithm. *Multimed Tools Appl* 80(1):1–39. <https://doi.org/10.1007/s11042-021-10909-w>
17. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. *Comput Vision Graph Image Process* 29(3):273–285. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)
18. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J Glob Optim* 39(3):459–471. <https://doi.org/10.1007/s10898-007-9149-x>
19. Kathirolu P, Kanmani S (2021) An efficient cluster-based routing using sparrow search algorithm for heterogeneous nodes in wireless sensor networks. In: *International Conference on Communication information and Computing Technology*. <https://doi.org/10.1109/ICCICT50803.2021.9510032>
20. Kennedy J, Eberhart R (2002) Particle swarm optimization. In: *Icnn95-international conference on neural networks*. IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
21. Khairuzzaman A, Chaudhury S (2020) Modified moth-flame optimization algorithm-based multilevel minimum cross entropy thresholding for image segmentation. *International Journal of Swarm Intelligence Research (IJSIR)* 11(4):123–139. <https://doi.org/10.4018/IJSIR.2020100106>

22. Kittler J, Illingworth J (1986) Minimum error thresholding. *Pattern Recogn* 19(1):41–47. [https://doi.org/10.1016/0031-3203\(86\)-90030-0](https://doi.org/10.1016/0031-3203(86)-90030-0)
23. Lei B, Fan J (2019) Image thresholding segmentation method based on minimum square rough entropy. *Appl Soft Comput* 84(1):105687. <https://doi.org/10.1016/j.asoc.2019.105687>
24. Li DQ (2021) A hybrid sparrow search algorithm. *Comput Knowl Technol* (05):232–234 (in China). <https://doi.org/10.14004/j.cnki.ckt.2021.0594>
25. Li CH, Lee CK (1993) Minimum cross entropy thresholding. *Pattern Recogn* 26(4):617–625. [https://doi.org/10.1016/0031-3203\(93\)90115-D](https://doi.org/10.1016/0031-3203(93)90115-D)
26. Li G, Zhao Y, Zhang L, Wang X et al (2020) Entropy-based global and local weight adaptive image segmentation models. *Tsinghua Sci Technol* 25(1):149–160. <https://doi.org/10.26599/TST.2019.9010026>
27. Liang J, Wang L, Ma M, Zhang J (2018) A fast Sar image segmentation method based on improved chicken swarm optimization algorithm. *Multimed Tools Appl* 77(24):31787–31805. <https://doi.org/10.1007/s11042-018-6119-x>
28. Liang Q, Chen B, Wu H, Ma C, Li S (2021) A novel modified sparrow search algorithm with application in side lobe level reduction of linear antenna array. *Wirel Commun Mob Comput* 2021(22):1–25. <https://doi.org/10.1155/2021/9915420>
29. Lin QQ, Zhang L, Wu TL, Mean TS, Tseng HW (2020) Application of tsallis cross-entropy in image thresholding segmentation. *Sensors Mater* 32(8):2771. <https://doi.org/10.18494/SAM.2020.2798>
30. Liu Y, Li S (2010) Two-dimensional arimoto entropy image thresholding based on ellipsoid region search strategy. In: *Multimedia Technology (ICMT), 2010 International conference on*. IEEE. <https://doi.org/10.1109/ICMULT.2010.5631047>
31. Liu JZ, Li WQ (1993) The automatic thresholding of gray-level pictures via two-dimensional Otsu method. *Acta Automat Sin* 19(1):101–105. <https://doi.org/10.3969/j.issn.1001-3695.2011.03.106>
32. Liu G, Shu C, Liang Z, Peng B, Cheng L (2021) A modified sparrow search algorithm with application in 3d route planning for UAV. *Sensors* 21(4):1224. <https://doi.org/10.3390/s21041224>
33. Lv X, Mu XD, Zhang J (2021) Multi threshold image segmentation based on improved sparrow search algorithm. *Syst Eng Electron* (02):318–327 (in China). <https://doi.org/10.12305/j.issn.1001-506x.2021.02.05>
34. Mahajan S, Mittal N, Pandit AK (2021) Image segmentation using multilevel thresholding based on type ii fuzzy entropy and marine predators algorithm. *Multimed Tools Appl* 80(13):19335–19359. <https://doi.org/10.1007/s11042-021-10641-5>
35. Mao QH, Zhang Q (2021) Improved sparrow algorithm integrating Cauchy mutation and reverse learning. *Comput Sci Explor* 15(6):1155–1164 (in China). <https://doi.org/10.3778/j.issn.1673-9418.2010032>
36. Meng X, Yu L, Gao X, Zhang H (2014) A new bio-inspired algorithm: chicken swarm optimization. In: *International conference in swarm intelligence*. Springer International Publishing. https://doi.org/10.1007/978-3-319-11857-4_10
37. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
38. Mohammadi-Balani A, Nayeri MD, Azar A, Taghizadeh-Yazdi M (2020) Golden eagle optimizer: a nature-inspired metaheuristic algorithm. *Comput Ind Eng* 152:107050. <https://doi.org/10.1016/j.cie.2020.107050>
39. Moussa M, Ouni HE, Douik A (2020) Edge detection based on fuzzy logic and hybrid types of shannon entropy. *J Circuits Syst Comput* 29(14):2050227. <https://doi.org/10.1142/S0218126620502278>
40. Nejad MR, Fartash M (2016) Applying chaotic imperialist competitive algorithm for multi-level image thresholding based on Kapur's entropy. *Adv Sci Technol Res J* 10(29):125–131. <https://doi.org/10.12913/22998624/61940>
41. Otsu N (2007) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
42. Pal NR, Pal SK (1989) Object-background segmentation using new definitions of entropy. *IEEE Proceedings Part E* 136(4):284–295. <https://doi.org/10.1049/ip-e.1989.0039>
43. Pal NR, Pal SK (1989) Entropic thresholding. *Signal Process* 16(2):97–108. [https://doi.org/10.1016/0165-1684\(89\)90090-X](https://doi.org/10.1016/0165-1684(89)90090-X)
44. Pan WT (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl-Based Syst* 26(2):69–74. <https://doi.org/10.1016/j.knosys.2011.07.001>
45. Peng C, Zou J, Liang M, Zhang Z, Xu H, Zhang Z (2020) A novel linear ccd variable velocity two-dimensional imaging approach based on adaptive image segmentation. *IEEE/ASME Trans Mechatron* 25(2):762–769. <https://doi.org/10.1109/TMECH.2020.2975096>
46. Qais MH, Hasanien HM, Alghuwainem S (2020) Transient search optimization: a new meta-heuristic optimization algorithm. *Appl Intell* 50(11):3926–3941. <https://doi.org/10.1007/s10489-020-01727-y>

47. Roy PC, Boddeti VN (2019) Mitigating information leakage in image representations: a maximum entropy approach. In: IEEE/CVF Conference on Computer Vision & Pattern Recognition. <https://doi.org/10.1109/CVPR.2019.00269>
48. Shen YX, Zhang XF et al (2021) Multiscale sine cosine optimization algorithm. *Control Decis*:1–10 (in China. <https://doi.org/10.13195/j.kzyjc.2021.0513>
49. Sm A, Smm B, Al A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
50. Takeda H, Yoshida S, Muneyasu M (2020) Learning from Noisy Labeled Data Using Symmetric Cross-Entropy Loss for Image Classification. In: 2020 IEEE 9th global conference on consumer electronics (GCCE). IEEE. <https://doi.org/10.1109/GCCE50665.2020.9291873>
51. Taruna AHD, Tiwari P (2021) A new parametric generalized exponential entropy measure on intuitionistic vague sets. *Int J Inf Technol* 13(4):1375–1380. <https://doi.org/10.1007/s41870-021-00655-5>
52. Tuerxun W, Xu C, Guo H, Jin Z, Zhou H (2021) Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm. *IEEE Access* 9:69307–69315. <https://doi.org/10.1109/ACCESS.2021.3075547>
53. Wang P, Zhang Y, Yang H (2021) Research on economic optimization of microgrid cluster based on chaos sparrow search algorithm. *Comput Intell Neurosci* 2021(3):1–18. <https://doi.org/10.1155/2021/5556780>
54. Wets JB (1981) Minimization by random search techniques. *Math Oper Res* 6(1):19–30. <https://doi.org/10.2307/3689263>
55. Wu DH, Kong F, Ji ZC (2017) Convergence analysis of chicken swarm algorithm. *Journal of Central South University: Natural Science Edition* 48(8):2105–2112. (in China. <https://doi.org/10.11817/j.issn.1672-7207.2017.08.018>
56. Xue JK, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering An Open Access Journal* 8(1):22–34. <https://doi.org/10.1080/21642583.2019.1708830>
57. Yan Z, Zhang J, Yang Z, Tang J (2020) Kapur's entropy for underwater multilevel thresholding image segmentation based on whale optimization algorithm. *IEEE Access* 9:41294–41319. <https://doi.org/10.1109/ACCESS.2020.3005452>
58. Yuan J, Zhao Z, Liu Y, He B, Gao Y (2021) Dmppt control of photovoltaic microgrid based on improved sparrow search algorithm. *IEEE Access* 9:16623–16629. <https://doi.org/10.1109/ACCESS.2021.3052960>
59. Zhang J, Xia K, He Z, Yin Z, Wang S (2021) Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection. *Math Probl Eng* 2021:1–18. <https://doi.org/10.1155/2021/6622935>
60. Zhou S, Xie H, Zhang C, Hua Y, Sui X (2021) Wavefront-shaping focusing based on a modified sparrow search algorithm. *Optik - International Journal for Light and Electron Optics* 244(35):167516. <https://doi.org/10.1016/j.ijleo.2021.167516>

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