Available online at https://jcupt.bupt.edu.cn

The Journal of China Universities of Posts and Telecommunications

April 2025, 32(2): 31 – 43

Research on Kapur multi-threshold image segmentation based on improved sparrow search algorithm

Wu Jin(✉), Feng Haoran, Chong Gege, Xiong Hao

School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Abstract

Multilevel threshold image segmentation divides an image into several regions with distinct characteristics. While effective, its computational complexity increases exponentially with the number of thresholds, highlighting the need for more efficient and stable methods. An improved sparrow search algorithm (ISSA) that combines multiple strategies to address the dependency on the initial population and solution accuracy issues in the basic sparrow search algorithm (SSA) was proposed in this paper. ISSA leverages circle chaotic mapping to enhance population diversity, a tangent flight operator to improve search diversity, and a triangular random walk to perturb the optimal solution, thereby enhancing global search capability and avoiding local optima. Performance evaluations on 16 benchmark functions demonstrate that ISSA surpasses the gray wolf optimizer (GWO), whale optimization algorithm (WOA), rat swarm optimizer (RSO), moth-flame optimization (MFO), and SSA in terms of search speed, accuracy, and robustness. When applied to multilevel threshold image segmentation, ISSA excels in Kapur's maximum entropy, peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and feature similarity (FSIM), highlighting its significant research value and application potential in the field of image segmentation.

Keywords image segmentation, sparrow search algorithm (SSA), multi-threshold, Kapur's maximum entropy

1 Introduction

Image segmentation is a fundamental technology in image processing, pattern recognition, and artificial intelligence, serving a crucial role in enhancing computer vision systems' understanding of images[1]. This technique divides an image into several independent regions with similar characteristics by analyzing attributes such as color, texture, grayscale, and edges, thus effectively distinguishing regions with differing traits. Within these regions, pixel values are consistent, whereas significant grayscale differences exist between regions. Researchers develop various segmentation algorithms, particularly those based on image attribute thresholds, such as inter-class variance, entropy, and moments[2]. These methods include, but are not limited to, Otsu's method, Kapur's maximum entropy method, and the minimum symmetric cross-entropy method.

The threshold-based segmentation method holds a significant position in traditional image segmentation techniques due to its simplicity, high efficiency, and broad applicability. This method categorizes images into two main categories: single-threshold and multi-

Special Issue: The 27th Annual Meeting of The China Association for Science and Technology

Corresponding author: Wu Jin, E-mail: wujin1026@126.com

DOI: 10.19682/j.cnki.1005-8885.2025.0016

threshold, based on the number of selected thresholds[3]. Compared to single-threshold segmentation, multi-threshold segmentation is more suitable for processing color images that are rich in information. However, as the number of thresholds increases, the computational load of traditional enumeration methods escalates rapidly, lengthening the segmentation time and potentially compromising result accuracy. Therefore, improving the efficiency and accuracy of multi-threshold segmentation is an important research direction. In recent years, the development of intelligent optimization algorithms has provided new pathways to address this issue. Researchers have explored integrating swarm intelligence optimization algorithms with traditional multi-threshold segmentation techniques to reduce computational complexity. Lü et al.[4] improved the position update strategies for explorers and followers in SSA, proposed an improved SSA that integrates bird flocking behavior optimization. The results show that this algorithm significantly enhanced speed, stability, and optimization accuracy in multi-threshold image segmentation. Zhao et al.[5] combined piecewise linear chaotic mapping and the firefly algorithm (FA) to improve the SSA, enhancing its ability to explore global optimal solutions. This improved algorithm is employed to optimize a backpropagation (BP) neural network model, successfully predicting PM2.5 concentrations in Xi'an. Jia et al.[6] applied the emerging grasshopper optimization algorithm (GOA) to image segmentation, demonstrating that this algorithm not only increased computational speed but also improved segmentation quality. Gao et al.[7] optimized the artificial bee colony (ABC) algorithm by adjusting adaptive parameters to automatically regulate the individuals' search steps. This improvement effectively addresses the high computational load and low efficiency of the original algorithm while boosting segmentation accuracy and convergence speed. Liu et al.[8] proposed an improved firefly algorithm (IFA) based on adaptive parameter control, addressing the issues of premature convergence and oscillation in traditional algorithms. They applied it to optimize the maximum entropy image segmentation algorithm, enhancing the accuracy and efficiency of image segmentation. Mishra et al.[9] proposed a multi-threshold image segmentation method based on WOA and social group optimization for brain magnetic resonance imaging (MRI) image segmentation, using Kapur's entropy as the objective function. The study indicated that this method outperformed particle swarm optimization (PSO)[10] and genetic algorithm (GA)[11] in robustness, fast convergence, and practical applicability. Experiments also point out that the segmentation method based on Otsu's method has a faster convergence speed compared to the one based on Kapur's entropy. Xue et al.[12] proposed the SSA, a new swarm intelligence optimization algorithm inspired by the foraging and anti-predation behaviors of sparrows. Compared to methods like PSO and GWO[13], SSA exhibits superior optimization performance. However, it has a longer runtime and occasionally became trapped in local optima. To address these shortcomings of the SSA, ISSA was proposed in this paper.

2 Principle of SSA

In nature, sparrows are social birds renowned for their intelligence and remarkable memory. Within their groups, they have clear divisions of labor: some sparrows are responsible for finding food and providing foraging areas and directions for the entire group, while the rest use this information to gather food. Additionally, when a sparrow senses danger, it promptly emits alarm signals, causing the whole flock to react quickly and evade predators. In the SSA, each sparrow's position represents a potential solution. During foraging, sparrows exhibit three key behaviors: acting as explorers to search for food, acting as followers follow the explorers to search for food, acting as scouts to determine whether to abandon the current food source. Explorers and followers can interchange roles, but their proportions remain constant. Explorers lead the foraging process by conducting extensive searches and continually updating their positions based on memory to locate food sources. Followers, on the other hand, improve their fitness by following the

explorers to forage. Due to the ever-present threat of predators, 10% to 20% of the sparrows in the population are randomly selected as scouts for surveillance. Upon detecting a predator, scouts immediately issue warnings, prompting the entire group to make anti-predator responses.

Explorers, update their positions according to

$$\boldsymbol{X}_{\mathrm{E}_{i,d}}^{t+1}=\begin{cases}\boldsymbol{X}_{\mathrm{E}_{i,d}}^{t}\mathrm{e}^{\frac{-i}{\alpha I_{\max}}}; & R_2<S_{\mathrm{T}}\\ \boldsymbol{X}_{\mathrm{E}_{i,d}}^{t}+Q\boldsymbol{E}; & R_2\geqslant S_{\mathrm{T}}\end{cases}\tag{1}$$

where t represents the current iteration number, $d=1,2,\ldots,D$. $\boldsymbol{X}_{\mathrm{E}_{i,d}}^{t}$ denotes the value of the dth dimension of the ith sparrow at the tth iteration. $I_{\max}$ is the maximum number of iterations. α is a random number between $(0,1]$. R_2 is a random number between $[0,1]$ and S_{T} is a constant between $[0.5,1]$ respectively represent alarm and safety thresholds. Q is a random number following a normal distribution. $\boldsymbol{E}$ represents a matrix where each element is 1, and D is the maximum dimension value. $R_2<S_{\mathrm{T}}$ means that there are no predators around, the explorers enters the wide search mode. $R_2\geqslant S_{\mathrm{T}}$ means that some sparrows have discovered the predator, and all sparrows need to quickly fly to other safe areas.

Followers update their positions using

$$\boldsymbol{X}_{\mathrm{F}_{i,d}}^{t+1}=\begin{cases}Q\mathrm{e}^{\frac{\boldsymbol{x}_{\mathrm{worst}}^{t}-\boldsymbol{X}_{\mathrm{F}_{i,d}}^{t}}{i^2}}; & i>\dfrac{n}{2}\\ \boldsymbol{X}_{\mathrm{P}}^{t+1}+|\boldsymbol{X}_{\mathrm{F}_{i,d}}^{t}-\boldsymbol{X}_{\mathrm{P}}^{t+1}|\boldsymbol{A}^{+}\boldsymbol{E}; & i\leqslant\dfrac{n}{2}\end{cases}\tag{2}$$

where $\boldsymbol{X}_{\mathrm{P}}^{t+1}$ is the best position occupied by the discoverer. $\boldsymbol{X}_{\mathrm{worst}}^{t}$ represents the current global worst position. $\boldsymbol{A}$ denotes a $1\times d$ matrix, where each element is randomly assigned as 1 or -1, and $\boldsymbol{A}^{+}=\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}$. When $i>n/2$, it suggests that the ith followers with the worse fitness value is most likely to be starving, n denotes the size of the population.

Scouts account for 10% – 20% of the total population. The initial positions of these sparrows are randomly generated within the population. Scouts update their positions using

$$\boldsymbol{X}_{\mathrm{S}_{i,d}}^{t+1}=\begin{cases}\boldsymbol{X}_{\mathrm{best}}^{t}+\beta|\boldsymbol{X}_{\mathrm{S}_{i,d}}^{t}-\boldsymbol{X}_{\mathrm{best}}^{t}|; & f_{\mathrm{i}}>f_{\mathrm{g}}\\ \boldsymbol{X}_{\mathrm{S}_{i,d}}^{t}+K\dfrac{|\boldsymbol{X}_{\mathrm{S}_{i,d}}^{t}-\boldsymbol{X}_{\mathrm{worst}}^{t}|}{(f_{\mathrm{i}}-f_{\mathrm{w}})+\varepsilon}; & f_{\mathrm{i}}=f_{\mathrm{g}}\end{cases}\tag{3}$$

where $\boldsymbol{X}_{\mathrm{best}}^{t}$ is the global best position at the tth iteration, β is a step size control parameter that follows a normal distribution with a mean of 0 and a variance of 1. K is a random value in the interval $[-1,1]$; f_{i}, f_{g}, and f_{w} represent the fitness of the current sparrow, the global best fitness, and the global worst fitness, respectively. To prevent the denominator from becoming zero, a very small constant ε is introduced as a substitute. For simplicity, when $f_{\mathrm{i}}>f_{\mathrm{g}}$, it indicates that the sparrow is at the edge of the group. X_{best}^{t} represents the position of the population center, where it is safe around it. When $f_{\mathrm{i}}=f_{\mathrm{g}}$, it means that the sparrow in the middle of the population senses danger and needs to move closer to the other sparrows.

3 Principle of ISSA

3.1 Circle chaotic mapping initialization

The performance of swarm intelligence optimization algorithms is largely influenced by the initialization of the initial population. In the traditional SSA, a random method is used to generate the initial population, which can sometimes lead to poor population quality, insufficient diversity, and excessive randomness. These factors can directly affect the algorithm's ability to perform global optimization searches in later stages. To improve this, chaotic variables, known for their comprehensiveness, randomness, and uniform distribution characteristics are introduced to optimize the initial solutions, thereby improving the quality of the initial population.

The ISSA employs circle chaotic mapping to enhance the population initialization process. The corresponding expression is

$$\boldsymbol{x}_{d+1}=\boldsymbol{x}_{d}+0.2-\mathrm{mod}\left(\frac{0.5}{2\pi}\sin(2\pi\boldsymbol{x}_{d}),1\right)\tag{4}$$

where, $\boldsymbol{x}_{d+1}$ represents the position after mapping, $\boldsymbol{x}_{d}$ represents the original position of the target, and d denotes the dimension.

Fig. 1 illustrates the circle chaotic mapping curve. In Fig. 1, it can be seen that the circle chaotic mapping produces more uniformly distributed random values, which enhances the algorithm's speed in

finding the optimal solution while reducing the likelihood of becoming trapped in local optima.

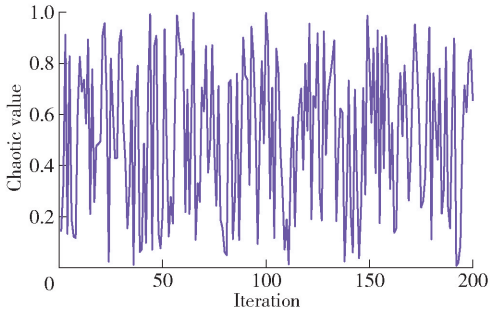

Fig. 1 Circle chaotic mapping curve

3.2 Tangent flight operator

The traditional SSA may be constrained by small step sizes and conservative movements during the search process, leading to limited exploration of the search space. This often leads to becoming trapped in local optima, especially in complex optimization problems. To address this, the tangent flight operator is applied to explorers. In each iteration, new step sizes are calculated using the tangent flight operator across different dimensions. These step sizes, combined with the current position and a randomly selected direction, are used to update the positions. Since the tangent function's output approaches infinity near $\pi/2$, it means that sometimes the step sizes can be very large. This characteristic allows the operator to potentially enables the operator to help the algorithm escape local optima and explore broader areas of the solution space. Conversely, because the tangent function's output changes slightly near 0, the operator can also perform small, precise searches within the solution space.

The calculation formula for the tangent flight operator can be expressed by

$$\boldsymbol{f} = \tan\left(\frac{\pi}{2}\boldsymbol{v}\right) \tag{5}$$

where, $\boldsymbol{v}$ is a $1 \times d$ row vector, where each of its elements follows a standard normal distribution (i.e., a normal distribution with mean 0 and variance 1), d is the dimension of the function. To facilitate an intuitive representation of the trajectory of the tangent flight operator, Fig. 2 shows the step size of the tangent flight operator's trajectory in a two-dimensional (2D) case. The x-axis represents the position in the first dimension of the search space, and the y-axis represents the position in the second dimension of the search space. The number of iterations is 200. In this context, the values on the x-axis and y-axis are dimensionless quantities, representing only the magnitude of step sizes.

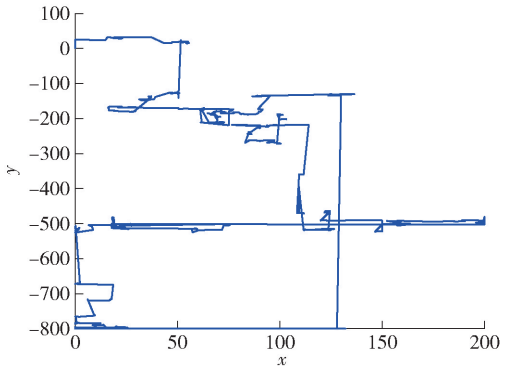

Fig. 2 Tangent flight operator trajectory diagram

The position update for explorers using the tangent flight operator is

$$\boldsymbol{X}_{\mathrm{E}_i}^{t+1} = \begin{cases} \boldsymbol{X}_{\mathrm{E}_i}^{t} \boldsymbol{f} \mathrm{e}^{\frac{-i}{\alpha T_{\max}}}; & R_2 < S_{\mathrm{T}} \\ \boldsymbol{X}_{\mathrm{E}_i}^{t} + \boldsymbol{Q}\boldsymbol{E}; & R_2 \geqslant S_{\mathrm{T}} \end{cases} \tag{6}$$

In Eq. (6), $R_2 < S_{\mathrm{T}}$ indicates the absence of predators, and the addition of the tangent flight operator can increase the diversity of the explorers' search. The explorers can move in various directions with different step sizes, enabling a more thorough exploration of the search space. This enhances the algorithm's global search capability and robustness.

3.3 Triangular random walk strategy

The random walk strategy is a statistical model used to describe random movement paths and represent irregular motion trajectories[14]. In traditional SSA, the search steps lack sufficient randomness, making it difficult to escape local optima. The triangular random walk strategy enables the algorithm to break free of local optima by varying both the search direction and step size. This strategy dynamically adjusts, increasing the algorithm's adaptability and flexibility. Applying this strategy to scouts allows for extensive exploration of

the solution space while enabling precise, in-depth exploitation of already discovered promising areas.

The triangular random walk process can be expressed by

$$r_1 = 0.1 - \frac{0.1t}{I_{\max}} \tag{7}$$

$$r_3 = r_2 r_1 \tag{8}$$

$$\boldsymbol{L}_{\mathrm{N}} = \boldsymbol{X}_{\mathrm{best}}^t - \boldsymbol{X}_i^t \tag{9}$$

$$\boldsymbol{L}_{\mathrm{P}} = \boldsymbol{L}_{\mathrm{N}} \boldsymbol{r}_4 \tag{10}$$

$$\boldsymbol{\alpha}_1 = \boldsymbol{L}_{\mathrm{N}}^2 + \boldsymbol{L}_{\mathrm{N}}^2 - 2\boldsymbol{L}_{\mathrm{P}}\boldsymbol{L}_{\mathrm{N}}\cos(2\pi \boldsymbol{r}_4) \tag{11}$$

The position update for scouts incorporating the triangular random walk strategy is

$$P_{\mathrm{d}} = 1 - \frac{t}{I_{\max}} \tag{12}$$

$$\boldsymbol{X}_{\mathrm{S}_i}^{t+1} = \begin{cases} \boldsymbol{X}_{\mathrm{best}}^t + r_3 \boldsymbol{\alpha}_1; & P_{\mathrm{d}} < r_2 \\ \boldsymbol{X}_{\mathrm{best}}^t + \beta |\boldsymbol{X}_i^t - \boldsymbol{X}_{\mathrm{best}}^t|; & f_{\mathrm{i}} > f_{\mathrm{g}} \\ \boldsymbol{X}_{\mathrm{S}_i}^t + K \dfrac{|\boldsymbol{X}_i^t - \boldsymbol{X}_{\mathrm{worst}}^t|}{f_{\mathrm{i}} - f_{\mathrm{g}} + \varepsilon}; & f_{\mathrm{i}} = f_{\mathrm{g}} \end{cases} \tag{13}$$

In Eq. (7) – Eq. (13), r_2 represents a random number within the range $[0,1]$, and $\boldsymbol{r}_4$ denotes a $1 \times d$ row vector with each element being a random number within $[0,1]$ interval. The algorithm in Eq. (9) computes the difference $\boldsymbol{L}_{\mathrm{N}}$ between the current position and the global best position, signifying the Euclidean distance to the global best position. By introducing randomness to $\boldsymbol{L}_{\mathrm{N}}$, the algorithm simulates two sides of a triangle, as shown in Eq. (10), with $\boldsymbol{L}_{\mathrm{P}}$ representing the random direction from the current position to the new position. Utilizing vectors $\boldsymbol{L}_{\mathrm{N}}$ and $\boldsymbol{L}_{\mathrm{P}}$, a random triangle is constructed. The algorithm forms the side length $\boldsymbol{\alpha}_1$ of this random triangle by calculating the cosine of the angle between these two vectors. In Eq. (12), P_{d} is characterized as a nonlinear probability perturbation factor that decreases from 1 to 0. When P_{d} is less than a random value within the range $[0,1]$, the triangular probability perturbation strategy is executed. The new position of an individual is randomly selected within a triangle with $\boldsymbol{X}_{\mathrm{best}}^t$ as the vertex. The new position is obtained by adding the step size and the triangle side length to the current best position. The step size r_3 dynamically decreases over time t, indicating a gradual reduction in the individual's range of exploration as the algorithm progresses. Through the triangular random walk strategy, the algorithm seeks a balance between global exploration and local exploitation, gradually reducing the randomness and range of the walk to focus the algorithm on promising regions.

4 Multi-threshold image segmentation based on ISSA

4.1 Kapur threshold segmentation principle

Kapur and others introduced the maximum entropy method in 1985, a technique for determining segmentation thresholds based on entropy. This method employs a multi-threshold approach for image segmentation, allowing for more precise identification of the foreground and background by dividing the image into multiple sections. The core principle of this method is based on Kapur entropy, which optimizes the segmentation effect by selecting thresholds that divide the image into different regions. The principle of multi-threshold image segmentation based on Kapur is as follows[9].

For single threshold segmentation, L is the maximum value of gray level, which is taken as 256 in this paper. if the threshold V divides the image into the target and background, the image entropy $H(V)$ can be expressed as $H(V) = H_0 + H_1$, here

$$H_0 = -\sum_{v=0}^{V} \frac{P_v}{\omega_0} \ln \frac{P_v}{\omega_0}; \quad \omega_0 = \sum_{v=0}^{V} P_v \tag{14}$$

$$H_1 = -\sum_{v=V+1}^{L-1} \frac{P_v}{\omega_1} \ln \frac{P_v}{\omega_1}; \quad \omega_1 = \sum_{v=V+1}^{L-1} P_v \tag{15}$$

Extended to multi-threshold, suppose the number of thresholds is q $(q > 1)$. the entropy of the image can be expressed as $H(V_1, V_2, \ldots, V_q) = H_0 + H_1 + \ldots + H_q$.

$$H_q = -\sum_{v=V_q+1}^{L-1} \frac{P_v}{\omega_q} \ln \frac{P_v}{\omega_q}; \quad \omega_q = \sum_{v=V_q+1}^{L-1} P_v \tag{16}$$

At this point, the Kapur entropy of the image grayscale values is defined as

$$f(V_1, V_2, \ldots, V_q) = H_0 + H_1 + \ldots + H_q \tag{17}$$

In Eqs. (15) – (17), ω_q represents the probability of each class of pixels, P_v represents the probability of

a pixel with grayscale value v appearing. $f(V_1, V_2, \ldots, V_q)$ represents the maximum entropy obtained by this set of thresholds. The threshold combination when the total entropy of the image is maximized can be expressed as

$$H(V_1^*, V_2^*, \ldots, V_q^*) = \arg \max_{0 \leqslant V_1 \leqslant V_2 \leqslant \ldots \leqslant V_q} H(V_1, V_2, \ldots, V_q) \quad (18)$$

4.2 Multi-threshold segmentation algorithm based on ISSA

If the image undergoes D-dimensional threshold segmentation, the solution vector for threshold segmentation is $\boldsymbol{T} = (V_1, V_2, \ldots, V_q)$. Each element in the solution vector is a positive integer that satisfies $0 < V_1 < V_2 < V_q < L$. Using Kapur entropy as the segmentation criterion, ISSA is applied to optimize within the L-level grayscale solution space. Once a solution vector that satisfies Eq. (18) is found, these solution vectors are considered optimal. Fig. 3 shows the process of multi-threshold image segmentation based on ISSA, with the specific steps as follows[15].

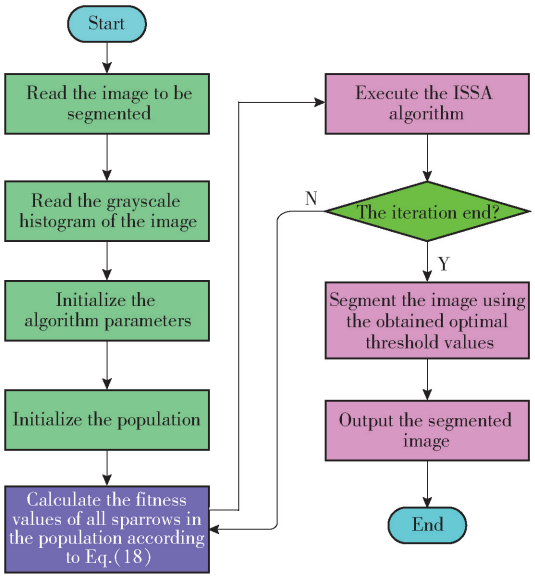

Fig. 3 Segmentation algorithm flowchart

Step 1 Read the color image and convert it into a grayscale histogram.

Step 2 Initialize the population. Introduce the circle chaotic mapping strategy and use Eq. (4) to initialize the position vectors of the sparrow population. Rank the fitness values of the sparrow individuals.

Step 3 Calculate the fitness value of each sparrow in the population according to Eq. (18), then update and save the position of the individual with the optimal fitness value

Step 4 Perform iterative calculations.

Step 5 Use the tangent flight operator and triangular random walk strategy to update the optimal sparrow.

Step 6 Check whether the termination condition of the iteration is met. If satisfied, exit the loop, and treat the global optimal solution obtained as the optimal threshold vector. Otherwise, jump to Step 4.

Step 7 When the termination condition is met, output the optimal threshold vector based on the position of the optimal solution and perform multi-threshold segmentation.

5 Experimental results and analysis

5.1 Benchmark function testing

To validate the effectiveness and accuracy of ISSA, comparisons were made by testing 16 benchmark functions using the GWO, WOA[16], RSO[17], MFO[18], arctic puffin optimization (APO)[19], goose algorithm (GOOSE)[20], SSA, and ISSA. It is worth noting that the data in this table are all dimensionless data. To evaluate the performance of the ISSA in optimization search capabilities, this study implemented a series of measures to minimize the bias caused by the randomness of metaheuristic optimization algorithms, ensuring the reliability and fairness of the experiments. Specifically, detailed parameter settings for the compared algorithms are provided, with specific values listed in Table 1. The test functions and their parameter settings are shown in Table 2. In this experiment, functions $F_1 - F_7$ are unimodal functions, used to verify the algorithms' convergence speed and computational accuracy. Functions $F_8 - F_{13}$ are multimodal functions, and functions $F_{14} - F_{16}$ are low-dimensional multimodal functions, designed to test the algorithms' exploitation and exploration capabilities in the global optimization process.

Table 1 Parameter value of different algorithms

Algorithm	Parameter	Value
GWO[13]	Exploration and exploitation control parameter	[0,2]
WOA[16]	Sprial hunting constant	1
RSO[17]	Exploration control parameter	[1,5]
	Exploitation constant parameter	[0,2]
MFO[18]	Convergence constant	[-2,-1]
	Logarithmic spiral	0.75
SSA[12]	Explorers	0.2
	Scouts	0.1
	Safety threshold	0.8
ISSA	Explorers	0.2
	Scouts	0.1
	Safety threshold	0.8
APO[19]	Synergy factor	0.5
	Exploration and exploitation switching threshold	0.5
GOOSE[20]	Exploration weight	[0,2]
	Select probability	[0,1]
	Time of arrive object	[0,1]
	Time of arrive sound	[0,1]
	Weight stone	[5,25]

Table 2 Benchmark function

Function	Dimension	Range	f_{min}
F_1	30	[-100,100]	0
F_2	30	[-10,10]	0
F_3	30	[-100,100]	0
F_4	30	[-100,100]	0
F_5	30	[-30,30]	0
F_6	30	[-100,100]	0
F_7	30	[-1.28,1.28]	0
F_8	30	[-500,500]	-12 569.5
F_9	30	[-5.12,5.12]	0
F_{10}	30	[-32,32]	0
F_{11}	30	[-600,600]	0
F_{12}	30	[-50,50]	0
F_{13}	30	[-50,50]	0
F_{14}	2	[-65.536,65.536]	0.998
F_{15}	6	[0,1]	-3.322
F_{16}	4	[0,10]	-10.153 2

The computer configuration used for the experiments is an Intel ® Core™ i7-10870U CPU@2.20 GHz with 16 GB of RAM, running Windows 11, and the programming environment is Matlab 2021b. To ensure objectivity, each algorithm was iterated 200 times with a population size of 30. To further minimize errors, each algorithm was independently run 30 times on the benchmark functions.

Table 3 presents the test results for six algorithms, and Fig. 4 illustrates the convergence graphs of the benchmark functions. Among them, "Best" is the optimal value of the results after 30 independent runs, "Mean" is the average value of the results after 30 independent runs, and "Std" is the standard deviation of the results after 30 independent runs. From the analysis of Table 3 and Fig. 4, for unimodal functions F_1, F_2, and F_4, ISSA demonstrates excellent performance in terms of the average and standard deviation of the optimal values. Its average and standard deviation are surpass those of other algorithms by more than 30 orders of magnitude, indicating that ISSA offers high solving accuracy, strong robustness, and fast convergence. For F_3, F_5, and F_6, although the standard deviation of ISSA is similar to that of SSA, ISSA performs better in terms of mean and optimal values. In the tests of multimodal functions F_8, F_9, F_{10}, F_{11}, and F_{12}, ISSA achieves the best results in terms of the average and standard deviation of the optimal values. Especially for F_{12}, the results among the algorithms reveals that MFO has the poorest solving capability, with the largest deviation from the theoretical optimal value. For low-dimensional multimodal functions F_{14}, F_{15}, and F_{16}, ISSA also performs the best in terms of the average and standard deviation of the optimal values. Although SSA reaches the theoretical optimal value for F_{14} and F_{16}, ISSA still surpasses SSA by more than 10 times of magnitude in standard deviation. As shown in Fig. 4, the ISSA outperforms GWO, WOA, RSO, MFO, and the original SSA in optimization capability across different benchmark functions. By utilizing circle chaotic mapping for SSA population initialization and introducing the tangent flight operator and random triangular walk strategy, the ISSA effectively avoids falling into local optima and balances exploration and exploitation capabilities. In summary, ISSA demonstrates excellent performance in solving speed, stability, and accuracy.

Table 3 Results of 16 benchmark functions for different algorithms

Function	Index	Results for different algorithms					
		ISSA	GWO	WOA	RSO	MFO	SSA
F_1	Best	0	$4.805\,7\times10^{-29}$	$1.154\,5\times10^{-86}$	0	$5.273\,0\times10^{-1}$	$6.256\,7\times10^{-258}$
	Mean	$3.139\,0\times10^{-112}$	$8.890\,6\times10^{-9}$	$8.839\,8\times10^{-27}$	$8.747\,5\times10^{-47}$	$2.886\,6\times10^{3}$	$1.257\,3\times10^{-37}$
	Std	$5.428\,2\times10^{-112}$	$1.008\,9\times10^{-8}$	$3.842\,5\times10^{-26}$	$4.764\,5\times10^{-46}$	$3.744\,9\times10^{3}$	$6.886\,3\times10^{-37}$
F_2	Best	0	$2.146\,4\times10^{-6}$	$2.285\,4\times10^{-23}$	0	$1.085\,17\times10^{1}$	$2.353\,7\times10^{-75}$
	Mean	$3.520\,3\times10^{-57}$	$5.588\,4\times10^{-6}$	$1.260\,6\times10^{-19}$	$3.017\,4\times10^{-26}$	$3.845\,7\times10^{1}$	$6.760\,7\times10^{-21}$
	Std	$6.782\,9\times10^{-57}$	$2.760\,5\times10^{-6}$	$4.724\,2\times10^{-19}$	$1.092\,3\times10^{-25}$	$1.835\,0\times10^{1}$	$2.169\,1\times10^{-20}$
F_3	Best	$4.715\,4\times10^{-118}$	$1.748\,1\times10^{-1}$	$2.855\,7\times10^{4}$	0	$8.572\,1\times10^{3}$	$7.495\,8\times10^{-83}$
	Mean	$2.656\,4\times10^{-22}$	$3.843\,3\times10^{0}$	$7.072\,6\times10^{4}$	$1.072\,1\times10^{5}$	$2.751\,1\times10^{4}$	$3.992\,9\times10^{-18}$
	Std	$1.455\,0\times10^{-21}$	$5.498\,9\times10^{0}$	$1.988\,4\times10^{4}$	$8.471\,6\times10^{4}$	$1.117\,0\times10^{4}$	$2.140\,1\times10^{-17}$
F_4	Best	0	$3.600\,0\times10^{-3}$	$1.030\,9\times10^{0}$	0	$5.667\,8\times10^{1}$	$1.814\,0\times10^{-82}$
	Mean	$5.533\,5\times10^{-57}$	$3.080\,0\times10^{-2}$	$5.651\,5\times10^{1}$	$3.280\,0\times10^{-26}$	$7.342\,8\times10^{1}$	$1.383\,2\times10^{-19}$
	Std	$1.319\,3\times10^{-56}$	$2.530\,0\times10^{-2}$	$2.904\,1\times10^{4}$	$1.127\,1\times10^{-25}$	$8.778\,2\times10^{0}$	$7.549\,3\times10^{-19}$
F_5	Best	$1.947\,0\times10^{-10}$	$2.637\,8\times10^{1}$	$2.815\,9\times10^{4}$	$2.864\,2\times10^{1}$	$7.825\,0\times10^{4}$	$7.729\,7\times10^{-7}$
	Mean	$1.600\,0\times10^{-3}$	$2.767\,9\times10^{1}$	$2.860\,8\times10^{1}$	$2.877\,8\times10^{1}$	$3.466\,1\times10^{6}$	$1.800\,0\times10^{-3}$
	Std	$2.600\,0\times10^{-3}$	$7.552\,0\times10^{-1}$	$1.871\,0\times10^{-1}$	$5.240\,0\times10^{-2}$	$1.120\,2\times10^{7}$	$2.400\,0\times10^{-3}$
F_6	Best	$6.412\,0\times10^{-10}$	$5.023\,0\times10^{-1}$	$4.237\,0\times10^{-1}$	$7.500\,0\times10^{0}$	$5.882\,1\times10^{2}$	$3.007\,6\times10^{-9}$
	Mean	$1.950\,8\times10^{-6}$	$1.345\,9\times10^{0}$	$1.184\,9\times10^{1}$	$7.500\,0\times10^{0}$	$2.389\,7\times10^{3}$	$3.258\,5\times10^{-6}$
	Std	$5.083\,4\times10^{-6}$	$4.577\,0\times10^{-1}$	$4.990\,0\times10^{-1}$	0	$3.096\,2\times10^{3}$	$6.455\,2\times10^{-6}$
F_7	Best	$3.840\,4\times10^{-5}$	$1.500\,0\times10^{-3}$	$3.200\,4\times10^{-5}$	$6.540\,0\times10^{-6}$	$3.383\,0\times10^{-1}$	$1.000\,0\times10^{-4}$
	Mean	$2.100\,0\times10^{-3}$	$5.600\,0\times10^{-3}$	$6.800\,0\times10^{-3}$	$2.965\,6\times10^{-4}$	$4.124\,0\times10^{0}$	$3.300\,0\times10^{-3}$
	Std	$3.600\,0\times10^{-3}$	$2.700\,0\times10^{-3}$	$9.200\,0\times10^{-3}$	$4.000\,0\times10^{-4}$	$1.177\,1\times10^{1}$	$2.500\,0\times10^{-3}$
F_8	Best	$-1.256\,9\times10^{4}$	$-7.659\,0\times10^{3}$	-1.256×10^{4}	$-7.614\,9\times10^{3}$	$-1.063\,0\times10^{4}$	$-9.638\,2\times10^{4}$
	Mean	$-1.256\,9\times10^{4}$	$-6.251\,0\times10^{3}$	$-9.748\,1\times10^{3}$	$-4.059\,1\times10^{3}$	$-8.623\,9\times10^{3}$	$-8.453\,1\times10^{4}$
	Std	$7.460\,0\times10^{-2}$	$6.880\,9\times10^{2}$	$1.641\,7\times10^{3}$	$1.664\,4\times10^{3}$	$8.150\,8\times10^{2}$	$6.965\,7\times10^{2}$
F_9	Best	0	$4.213\,5\times10^{0}$	0	0	$7.935\,6\times10^{1}$	0
	Mean	0	$1.276\,5\times10^{1}$	$1.515\,8\times10^{-14}$	0	$1.749\,8\times10^{2}$	0
	Std	0	$8.165\,7\times10^{0}$	$3.930\,7\times10^{-14}$	0	$3.844\,3\times10^{1}$	0
F_{10}	Best	$8.881\,8\times10^{-16}$	$8.987\,9\times10^{-6}$	$7.993\,6\times10^{-15}$	$8.881\,8\times10^{-16}$	$6.230\,4\times10^{0}$	$8.881\,8\times10^{-16}$
	Mean	$8.881\,8\times10^{-16}$	$1.688\,0\times10^{-5}$	$1.900\,7\times10^{-14}$	$8.881\,8\times10-16$	$1.705\,3\times10^{1}$	$1.006\,6\times10^{-15}$
	Std	0	$7.469\,3\times10^{-6}$	$1.147\,8\times10^{-14}$	0	$4.022\,8\times10^{0}$	$6.486\,3\times10^{-16}$
F_{11}	Best	0	$2.724\,3\times10^{-9}$	0	0	$5.801\,5\times10^{0}$	0
	Mean	0	$1.240\,0\times10^{-2}$	$4.650\,0\times10^{-2}$	0	$3.654\,3\times10^{1}$	0
	Std	0	$1.420\,0\times10^{-2}$	$1.460\,0\times10^{-1}$	0	$5.588\,9\times10^{1}$	0
F_{12}	Best	$1.320\,4\times10^{-11}$	$3.590\,0\times10^{-2}$	$1.610\,0\times10^{-2}$	$1.669\,0\times10^{0}$	$2.677\,0\times10^{3}$	$1.324\,7\times10^{-11}$
	Mean	$5.529\,0\times10^{-8}$	$1.039\,0\times10^{-1}$	$7.690\,0\times10^{-2}$	$1.669\,0\times10^{0}$	$5.621\,4\times10^{5}$	$1.475\,6\times10^{-7}$
	Std	$2.297\,0\times10^{-7}$	$6.760\,0\times10^{-2}$	$4.590\,0\times10^{-2}$	$1.129\,2\times10^{-15}$	$2.653\,1\times10^{7}$	$3.299\,6\times10^{-7}$
F_{13}	Best	$6.726\,0\times10^{-12}$	$4.496\,0\times10^{-1}$	$3.385\,0\times10^{-1}$	$8.630\,0\times10^{-1}$	$3.723\,1\times10^{4}$	$1.290\,2\times10^{-9}$
	Mean	$7.602\,8\times10^{-7}$	$1.088\,2\times10^{0}$	$9.953\,0\times10^{-1}$	$1.650\,7\times10^{0}$	$1.693\,8\times10^{6}$	$2.612\,9\times10^{-7}$
	Std	$1.208\,0\times10^{-6}$	$2.915\,0\times10^{-1}$	$3.855\,0\times10^{-1}$	$6.523\,0\times10^{-1}$	$2.280\,6\times10^{6}$	$3.605\,8\times10^{-7}$
F_{14}	Best	$9.980\,0\times10^{3}$	$9.980\,0\times10^{-1}$	$9.980\,0\times10^{-1}$	$2.992\,1\times10^{0}$	$9.980\,0\times10^{-1}$	$9.980\,0\times10^{-1}$
	Mean	$9.980\,0\times10^{3}$	$5.989\,7\times10^{0}$	$4.066\,9\times10^{0}$	$1.164\,8\times10^{1}$	$2.809\,1\times10^{0}$	$6.063\,8\times10^{0}$
	Std	$4.728\,3\times10^{-16}$	$4.329\,0\times10^{0}$	$4.060\,4\times10^{0}$	$2.439\,7\times10^{0}$	$2.055\,7\times10^{0}$	$5.541\,4\times10^{0}$
F_{15}	Best	$-3.322\,0\times10^{0}$	$-3.322\,0\times10^{0}$	$-3.321\,5\times10^{0}$	$-3.283\,0\times10^{0}$	$-3.322\,0\times10^{0}$	$-3.322\,0\times10^{0}$
	Mean	$-3.313\,9\times10^{0}$	$-3.266\,3\times10^{0}$	$-3.171\,8\times10^{0}$	$-2.495\,2\times10^{0}$	$-3.226\,3\times10^{0}$	$-3.270\,3\times10^{0}$
	Std	$3.020\,0\times10^{-2}$	$7.830\,0\times10^{-2}$	$1.559\,0\times10^{-1}$	$5.580\,0\times10^{-1}$	$5.770\,0\times10^{-2}$	$5.990\,0\times10^{-2}$
F_{16}	Best	$-1.015\,3\times10^{1}$	$-1.015\,1\times10^{1}$	$-1.013\,1\times10^{-1}$	$-5.334\,4\times10^{0}$	$-1.015\,1\times10^{1}$	$-1.015\,1\times10^{1}$
	Mean	$-1.015\,3\times10^{1}$	$-9.394\,6\times10^{0}$	$-7.136\,6\times10^{0}$	$-3.082\,4\times10^{0}$	$-7.891\,4\times10^{0}$	$-7.604\,2\times10^{0}$
	Std	$3.499\,3\times10^{-14}$	$2.282\,1\times10^{0}$	$2.922\,8\times10^{0}$	$1.067\,6\times10^{0}$	$3.112\,2\times10^{0}$	$2.592\,6\times10^{0}$

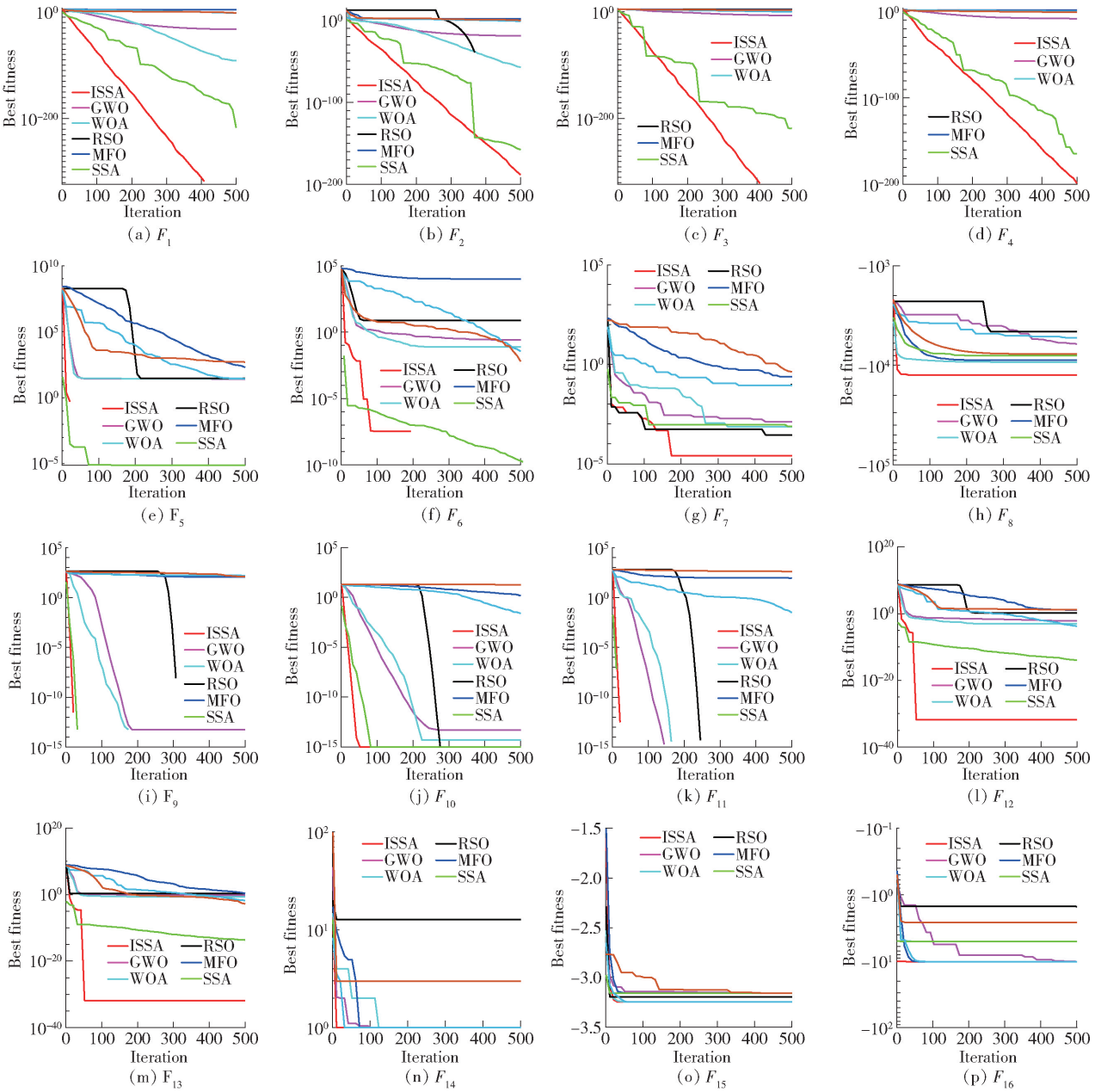

Fig. 4 Convergence curves of the best fitness values of different cdgorithms

5.2 Kapur multi-threshold image segmentation based on ISSA

To validate the feasibility of Kapur multi-threshold image segmentation based on ISSA, images from the ADE20K dataset (https://paperswithcode.com/sota) are selected as test images. The experiments involved 4-threshold, 6-threshold, and 8-threshold image segmentation based on the Kapur entropy segmentation criterion.

The algorithm parameters are set as follows: population size is 30, maximum iterations is 200, grayscale level of each image is 256, and segmentation threshold range is [0, 255]. The proposed ISSA combined with maximum entropy is used for image segmentation experiments. The segmentation performance is evaluated by comparing the fitness function value, PSNR[21], SSIM[22], and FSIM[23] with those of GWO, WOA, RSO, MFO, SSA, APO[19], and GOOSE[20].

Consider an image with a pixel value of $H \times W$. PSNR is used to verify the similarity between the original image I^{o} and the segmented image I^{s}. Its expression is

$$\gamma = \frac{1}{H_w} \sum_{h=1}^{H} \sum_{w=1}^{W} (I_{hw}^{o} - I_{hw}^{s})^2 \tag{19}$$

$$\eta = 10\lg \frac{255^2}{\gamma} \tag{20}$$

where γ is the mean square error (MSE) between the original image I^{o} and the processed image I^{s}, I_{hw}^{o} and I_{hw}^{s} represent the pixel values at the hth row and wth column of the corresponding images. η denotes the PSNR. SSIM δ is used to compare the SSIM between the original image and the segmented image, that is, the relationship between neighboring pixels in the image. Its expression is

$$\delta = \frac{(2\mu_o\mu_s + \lambda_1)(2\sigma_{os} + \lambda_2)}{(\mu_o^2 + \mu_s^2 + \lambda_1)(\sigma_o^2 + \sigma_s^2 + \lambda_2)} \tag{21}$$

where μ_o and μ_s are the mean pixel values of the original and segmented images, respectively. σ_o and σ_s are the standard deviations of the images, and σ_{os} represents the covariance between the pixels of the original image I^{o} and the segmented image I^{s}. λ_1 and λ_2 are constants. To avoid division by zero, it is common to set $\lambda_1 = (0.01L)^2$ and $\lambda_2 = (0.03L)^2$.

FSIM, ε, is used to compare the FSIM between images, specifically phase congruency (PC) and gradient magnitude (GM). The expression is

$$S_{PC} = \frac{2P_o(z)P_s(z) + \lambda_1}{P_o^2(z) + P_s^2(z) + \lambda_2} \tag{22}$$

$$S_G = \frac{2G_o(z)G_s(x) + \lambda_1}{G_o^2(x) + G_s^2(x) + \lambda_2} \tag{23}$$

$$S_L(z) = S_{PC}(x)S_G(x) \tag{24}$$

$$\varepsilon = \frac{\sum_{l=0}^{L-1} S_L(x)P_{max}(x)}{\sum_{l=0}^{L-1} P_{max}(x)} \tag{25}$$

where, $S_{PC}(x)$ denotes the PC at position x, where $P_o(x)$ and $P_s(x)$ are the phase values of the two images at position x, respectively. $S_G(x)$ represents the gradient similarity, where $G_o(x)$ and $G_s(x)$ are the gradient values of the two images at position x, respectively. The x in the formula usually represents the position coordinate or pixel index in the image, $P_{max}(x) = \max\{P_o(x), P_s(x)\}$.

PSNR, SSIM, and FSIM share a common characteristic: the larger the obtained value, the higher the quality of the image resulting from the segmentation. The test images are shown in Fig. 5. Table 4 presents the experimental results of ISSA and the other algorithms, GWO, WOA, RSO, MFO, SSA, APO, and GOOSE in image threshold segmentation, comparing the average maximum entropy fitness values obtained after 30 runs of each algorithm.

(a) Room

(b) Hill

(c) City

Fig. 5 Test images

Table 4 Maximum entropy value of the segmented image fitness

Image	Threshold	ISSA	GWO	WOA	RSO	MFO	SSA	APO	GOOSE
Room	4	18.076 2	18.055 4	18.055 6	17.642 2	18.056 2	18.056 2	18.056 2	17.643 4
	6	23.028 7	23.001 7	23.003 3	22.309 9	22.951 7	23.009 4	22.998 1	22.559 8
	8	27.704 0	27.635 8	27.668 0	26.766 1	27.630 2	27.680 2	27.540 7	27.332 3
Hill	4	18.368 2	18.374 5	18.379 7	18.016 1	18.364 0	18.360 8	18.400 8	18.107 6
	6	23.976 7	23.946 3	23.948 3	23.154 7	23.958 9	23.945 4	23.957 1	23.545 8
	8	28.949 1	28.891 6	28.892 1	27.775 7	28.968 6	28.965 7	28.823 4	28.620 8
City	4	19.438 7	19.417 5	19.418 5	19.127 8	19.418 7	19.418 7	19.418 7	19.196 7
	6	24.982 4	24.959 1	24.964 6	24.263 9	24.967 3	24.971 6	24.962 0	24.622 4
	8	29.932 1	29.889 7	29.905 6	28.671 4	29.912 9	29.918 9	29.825 9	29.751 7

Tables 5 – 7 present the average PSNR value, average SSIM value, and average FSIM value obtained after 30 runs of each algorithm, respectively.

Table 5 Average PSNR value of the segmented images dB

Image	Threshold	ISSA	GWO	WOA	RSO	MFO	SSA	APO	GOOSE
Room	4	18.873 6	18.861 3	18.869 9	17.554 1	18.869 1	18.866 5	18.873 1	18.582 0
	6	20.501 3	20.455 0	20.470 0	19.254 9	20.927 3	20.467 6	20.472 3	21.364 2
	8	25.279 3	25.041 2	25.146 9	22.426 1	24.785 0	25.087 6	24.826 4	24.081 3
Hill	4	20.762 9	19.767 1	19.557 2	19.650 4	20.429 3	20.476 4	19.557 9	19.611 8
	6	23.580 0	23.570 1	23.552 8	22.254 1	23.410 8	23.388 7	23.475 0	22.679 5
	8	26.350 9	25.646 8	25.661 0	23.408 4	25.869 1	25.929 3	25.288 5	25.231 4
City	4	20.413 2	20.398 1	20.412 1	20.138 6	20.406 3	20.406 3	20.409 6	19.775 8
	6	23.493 9	23.544 7	23.558 4	22.021 6	23.485 6	23.508 5	23.515 4	22.301 5
	8	25.901 7	25.726 8	25.888 5	23.563 1	25.780 3	25.835 6	25.714 5	25.250 1

Table 6 Average SSIM value of the segmented images

Image	Threshold	ISSA	GWO	WOA	RSO	MFO	SSA	APO	GOOSE
Room	4	0.746 5	0.745 9	0.743 7	0.723 1	0.746 4	0.745 9	0.746 3	0.693 3
	6	0.831 5	0.830 4	0.830 1	0.809 9	0.823 0	0.830 7	0.829 2	0.784 3
	8	0.836 6	0.839 2	0.841 7	0.841 7	0.841 3	0.835 0	0.828 6	0.846 4
Hill	4	0.797 9	0.760 2	0.757 5	0.783 1	0.772 5	0.775 6	0.745 6	0.772 7
	6	0.849 8	0.846 5	0.846 1	0.829 6	0.848 0	0.845 5	0.846 9	0.821 5
	8	0.878 1	0.872 7	0.872 2	0.840 7	0.874 4	0.874 4	0.868 8	0.867 6
City	4	0.675 7	0.673 8	0.673 9	0.673 4	0.673 7	0.673 7	0.673 5	0.645 9
	6	0.764 6	0.763 8	0.764 0	0.740 2	0.763 2	0.763 5	0.762 2	0.737 1
	8	0.823 3	0.816 0	0.818 1	0.784 8	0.815 8	0.821 6	0.818 7	0.804 3

Table 7 Average FSIM value of segmented images

Image	Threshold	ISSA	GWO	WOA	RSO	MFO	SSA	APO	GOOSE
Room	4	0.892 9	0.892 7	0.892 8	0.871 6	0.892 8	0.892 8	0.892 8	0.883 2
	6	0.923 4	0.919 1	0.919 5	0.900 5	0.919 5	0.919 6	0.919 2	0.922 1
	8	0.955 7	0.954 5	0.955 7	0.935 0	0.953 3	0.954 9	0.951 4	0.949 5
Hill	4	0.878 6	0.870 0	0.866 2	0.865 3	0.877 3	0.878 6	0.863 6	0.861 1
	6	0.929 2	0.928 2	0.927 1	0.895 6	0.927 9	0.927 2	0.929 0	0.910 4
	8	0.954 0	0.948 5	0.948 2	0.913 6	0.951 6	0.951 3	0.946 3	0.942 4
City	4	0.928 0	0.928 2	0.928 4	0.920 6	0.928 3	0.928 3	0.928 1	0.914 1
	6	0.963 2	0.962 6	0.962 5	0.938 0	0.962 5	0.962 6	0.963 0	0.946 6
	8	0.976 2	0.975 3	0.975 8	0.953 6	0.975 0	0.975 8	0.974 7	0.969 3

From the data in Tables 4 – 7, it is evident that the results based on the Kapur method show that ISSA achieves higher entropy values in 77.78% of the cases, demonstrating superior stability. In 88.89% of the cases, ISSA achieved higher PSNR values, in 88.89% of the cases, ISSA achieved the optimal SSIM values, and in 88.89% of the cases, ISSA achieved the optimal FSIM value. In summary, ISSA exhibits higher segmentation accuracy and quality, stronger algorithm stability, and the best overall segmentation performance in multi-threshold image segmentation applications.

To visually present the multi-threshold image segmentation results based on ISSA, Fig. 6 illustrates the image segmentation results of ISSA under the Kapur entropy multi-threshold segmentation criterion. As the number of thresholds increases, the segmentation results of ISSA exhibit clearer details, more comprehensive information, and higher segmentation quality.

(a) Room 4-threshold (b) Room 6-threshold (c) Room 8-threshold

(d) Hill 4-threshold (e) Hill 6-threshold (f) Hill 8-threshold

(g) City 4-threshold (h) City 6-threshold (i) City 8-threshold

Fig. 6 Kapur-based multi-threshold segmentation results

The fitness function values obtained from 30 independent runs of the Kapur method were subjected to a Wilcoxon signed-rank test with a significance level of 5%. The null hypothesis H_0 assumes no difference in the fitness function values between the two methods, while the alternative hypothesis H_1 assumes a significant difference in the fitness function values between the two methods. The test results are shown in Table 8, it can be seen from Table 8 that Wilcoxon rank-sum are less than 0.05 in 92.06% of the cases. Therefore, H_0 is rejected, and H_1 is accepted, indicating a significant difference in the Kapur entropy function value between the two methods, confirming that the results are statistically significant.

Table 8 Fitness value of Wilcoxon rank-sum test

Image	Threshold	GWO	WOA	RSO	MFO	SSA	APO	GOOSE
Room	4	$2.644\,4\times10^{-13}$	$2.038\,4\times10^{-12}$	$2.363\,8\times10^{-12}$	$4.157\,4\times10^{-14}$	$4.157\,4\times10^{-14}$	$4.157\,4\times10^{-14}$	$2.363\,8\times10^{-12}$
	6	$2.145\,8\times10^{-11}$	$2.246\,3\times10^{-11}$	$2.246\,3\times10^{-11}$	$1.747\,7\times10^{-11}$	$5.567\,1\times10^{-12}$	$2.166\,7\times10^{-11}$	$2.246\,3\times10^{-11}$
	8	$3.157\,1\times10^{-10}$	$5.569\,6\times10^{-10}$	$3.018\,0\times10^{-11}$	$2.037\,5\times10^{-10}$	$3.453\,4\times10^{-10}$	$4.972\,1\times10^{-11}$	$3.336\,3\times10^{-11}$
Hill	4	$5.365\,1\times10^{-1}$	$8.994\,1\times10^{-1}$	$2.319\,6\times10^{-11}$	$2.981\,0\times10^{-2}$	$1.559\,1\times10^{-2}$	$3.438\,3\times10^{-1}$	$1.263\,2\times10^{-10}$
	6	$7.064\,3\times10^{-9}$	$8.428\,7\times10^{-9}$	$2.993\,5\times10^{-11}$	$1.528\,9\times10^{-9}$	$3.033\,6\times10^{-9}$	$8.207\,4\times10^{-9}$	$9.836\,4\times10^{-11}$
	8	$2.510\,1\times10^{-2}$	$1.301\,7\times10^{-3}$	$3.019\,9\times10^{-11}$	$9.820\,6\times10^{-1}$	$8.647\,1\times10^{-1}$	$7.772\,6\times10^{-9}$	$9.063\,2\times10^{-8}$
City	4	$9.237\,2\times10^{-13}$	$2.694\,9\times10^{-12}$	$3.150\,7\times10^{-12}$	$6.115\,1\times10^{-14}$	$6.115\,1\times10^{-14}$	$6.115\,1\times10^{-14}$	$3.150\,7\times10^{-12}$
	6	$7.444\,3\times10^{-9}$	$7.607\,0\times10^{-9}$	$2.615\,3\times10^{-11}$	$8.897\,7\times10^{-10}$	$1.651\,3\times10^{-9}$	$6.627\,5\times10^{-9}$	$3.910\,6\times10^{-11}$
	8	$1.853\,5\times10^{-6}$	$2.868\,0\times10^{-6}$	$2.997\,2\times10^{-11}$	$1.911\,5\times10^{-4}$	$3.328\,4\times10^{-2}$	$1.166\,3\times10^{-9}$	$8.834\,1\times10^{-10}$

6 Conclusions and future work

Based on the SSA, circle chaotic initialization for the population, a tangent flight operator strategy, and a triangular random walk strategy to significantly improve the convergence accuracy of SSA and address its tendency to fall into local optima. Experimental results on 16 benchmark test functions demonstrate that the ISSA has faster convergence speed, higher accuracy, and greater robustness.

Furthermore, the ISSA is applied to image segmentation using Kapur as the objective function to enhance the quality of the segmented images. After determining the segmentation thresholds, four evaluation metrics, Kapur maximum entropy fitness function value, PSNR, SSIM, and FSIM are used to analyze the segmented images compared to the original

images. The results confirms the superiority of the algorithm.

Finally, a rank-sum test reveals a significant difference between the Kapur entropy function values of ISSA and the other algorithms, indicating non-randomness. Simulation experiments verifies the feasibility of ISSA for practical engineering applications.

The analysis of the ISSA's parameters in this paper is not yet comprehensive. Future research could further explore the impact of various parameters on the algorithm's performance and enhance its optimization capability by refining these parameters. Additionally, we will consider integrating the ISSA with the Otsu image segmentation algorithm and the minimum cross entropy image segmentation algorithm to evaluate their effectiveness in improving the accuracy and efficiency of image segmentation.

Acknowledgements

This work was supported by the National Key R&D Program of China: Science and Technology Innovation 2030 (2022ZD0119000).

References

[1] FU X. Research on multi-threshold image segmentation based on multi-strategy fusion improved swarm intelligence optimization. Master Thesis. Harbin, China: Northeast Forestry University, 2023 (in Chinese).

[2] LI M. Research on multilevel thresholding image segmentation based on swarm intelligence optimization algorithm. Master Thesis. Wuhan, China: Hubei University of Technology, 2021 (in Chinese).

[3] WU L S, CHENG W, HU Y. Image segmentation of multilevel threshold based on improved cuckoo search algorithm. Journal of Jilin University (Engineering and Technology Edition), 2021, 51(1): 358 – 369 (in Chinese).

[4] LÜ X, MU X D, ZHANG J. Multi-threshold image segmentation based on improved sparrow search algorithm. Systems Engineering and Electronics, 2021, 43(2): 318 – 327 (in Chinese).

[5] ZHAO K, SHI Y, NIU M J, et al. Prediction of PM2. 5 concentration based on optimized BP neural network with improved sparrow search algorithm. Journal of Surveying and Mapping, 2022, (10): 44 – 50 (in Chinese).

[6] JIA H M, LANG C B, OLIVA D, et al. Hybrid grasshopper optimization algorithm and differential evolution for multilevel satellite image segmentation. Remote Sensing, 2019, 11(9): Article 1134.

[7] GAO H, ZHENG F, PUN C M, et al. A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm. Computers & Electrical Engineering, 2018, 70: 931 – 938.

[8] LIU Q R, JIANG Z J, SHI H Q. Maximum entropy image segmentation method based on improved firefly algorithm. Journal of Physics: Conference Series, 2019, 1213(3): Article 032023.

[9] MISHRA P K, SATAPATHY S C, ROUT M. Multi-level Kapur's thresholding using whale optimization and social group optimization for brain MRI image segmentation. Journal of Information and Optimization Sciences, 2022, 43(5): 1039 – 1045.

[10] KENNEDY J, EBERHART R. Particle swarm optimization. Proceedings of the 1995 International Conference on Neural Networks (ICNN'95), 1995, Nov 27 – Dec 1, Perth, Australia. Piscataway, NJ, USA: IEEE, 1995: 1942 – 1948.

[11] HOLLAND J H. Genetic algorithms. Scientific American, 1992, 267(1): 66 – 73.

[12] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm. Systems Science & Control Engineering, 2020, 8(1): 22 – 34.

[13] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer. Advances in Engineering Software, 2014, 69: 46 – 61.

[14] SONG M J, JIA H M, ABUALIGAH L, et al. Modified Harris Hawks optimization algorithm with exploration factor and random walk strategy. Computational Intelligence and Neuroscience, 2022, DOI: 10.1155/2022/4673665.

[15] MA X J, HE H, WANG H W, et al. Maximum exponential entropy segmentation method based on improved sparrow search algorithm. Science Technology and Engineering, 2023, 23(16): 6983 – 6992 (in Chinese).

[16] MIRJALILI S, LEWIS A. The whale optimization algorithm. Advances in Engineering Software, 2016, 95: 51 – 67.

[17] DHIMAN G M, GARG A, KUMAR N V, et al. A novel algorithm for global optimization: rat swarm optimizer. Journal of Ambient Intelligence and Humanized Computing, 2021, 12: 8457 – 8482.

[18] MIRJALILI S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 2015, 89: 228 – 249.

[19] WANG W C, TIAN W C, XU D M, et al. Arctic puffin optimization: a bio-inspired metaheuristic algorithm for solving engineering design optimization. Advances in Engineering Software, 2024, 195: Article 103694.

[20] HAMAD R K, RASHID T A. GOOSE algorithm: a powerful optimization tool for real-world engineering challenges and beyond. Evolving Systems, 2024: 15(4): 1249 – 1274.

[21] ELSHAZLY E, ABDELWAHAB S, ABOUZAID R, et al. A secure image steganography algorithm based on least significant bit and integer wavelet transform. Journal of Systems Engineering and Electronics, 2018, 29(3): 639 – 649.

[22] TANG Y M, REN F J, PEDRYCZ W. Fuzzy C-means clustering through SSIM and patch for image segmentation. Applied Soft Computing, 2020, 87: Article 105928.

[23] TIAN J, LI Y X, LI T Y. Contrastive study of activation function in convolutional neural network. Computer Systems & Applications, 2018, 27(7): 43 – 49 (in Chinese).

(Editor: Wang Xuying)