ACCSCIENCE PUBLISHING
An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.16, No.2, pp.404-427 (2026)
https://doi.org/10.36922/IJOCTA025410176
IJOCTA
RESEARCH ARTICLE

The strategy-based fractional-order slime mould algorithm: Application in image segmentation

Ruiyang Wu, Yu Ma*, Bowen Xiao, and Shoutong Huang

Department of New-Generation Electronic Information Engineering, School of Electronic and Electrical Engineering, Ningxia University, Yinchuan, Ningxia, China
12023130627@stu.nxu.edu.cn, may@nxu.edu.cn, 12022130647@stu.nxu.edu.cn, 12024130722@stu.nxu.edu.cn

ARTICLE INFO

Article History:
Received: October 12, 2025
Revised: November 22, 2025
Accepted: November 27, 2025
Published Online: January 6, 2026

Keywords:
2-D Otsu
Fractional-order differentiation
Image segmentation
Slime Mould Algorithm

ABSTRACT

The traditional Slime Mould Algorithm (SMA) often suffers from slow convergence and a tendency to fall into local optima, significantly limiting its applicability in complex optimization problems. To overcome these limitations, a Strategy-based Fractional-order SMA (SFSMA) that integrates multi-strategy integration is proposed in this paper. Population diversity was enhanced through the synergistic combination of differential evolution and fractional-order calculus, while a dynamic threshold mechanism monitored the search state in real time. These strategies collectively facilitated escape from local optima and improved convergence speed. To evaluate the performance of the proposed method, SFSMA was first compared with mainstream swarm intelligence algorithms and other fractional-order swarm intelligence variants using 12 classical benchmark functions. Experimental results confirm that SFSMA achieved significant improvements in both convergence speed and optimization accuracy. Furthermore, an SFSMA–Otsu segmentation model was developed by integrating the proposed algorithm with the two-dimensional Otsu algorithm. The model was evaluated on multiple types of images, including human, landscape, and medical images, using four quantitative metrics: peak signal-to-noise ratio, mean squared error, structural similarity index measure, and feature similarity index measure. Quantitative results demonstrate that the SFSMA–Otsu achieved substantially higher segmentation accuracy compared to existing methods. In addition, the convergence speed of SFSMA improved by approximately 82.08% compared to the traditional SMA. In conclusion, the proposed SFSMA effectively addresses the shortcomings of traditional SMA and provides an efficient and reliable solution for complex optimization and image segmentation tasks, exhibiting both theoretical value and practical potential.

1. Introduction

In recent years, to reduce computational costs and enable automatic threshold selection, swarm intelligence algorithms have been used in image threshold segmentation. They provide an efficient way to optimize thresholds for complex image segmentation problems. Liu et al. [1] improved the Firefly Algorithm (FA) by combining a cell membrane optimization mechanism with an adaptive step size strategy. In multi-threshold Otsu image segmentation, the improved FA achieves faster convergence speed and better segmentation quality. Bhandari et al. [2] applied the Cuttlefish Algorithm to optimize three-dimensional Otsu, which effectively improved the segmentation effect of noisy or low-contrast images. Dong

*Corresponding Author

et al. [3] proposed an Improved Dragonfly Algorithm for Otsu multi-threshold image segmentation. They introduced an adaptive distribution strategy to balance global–local search and accelerate convergence. Ma et al. [4] proposed the Hybrid Fractional-order Butterfly Optimization Algorithm (HFBOA), which effectively improved the accuracy and efficiency of image segmentation. Wang et al. [5] developed a Mixed-strategy-based improved Whale Optimization Algorithm (WOA), which effectively overcomes the limitation of the traditional WOA. The proposed method enables efficient, precise grayscale image segmentation through accurate, optimal multilevel threshold search. Sowjanya and Injeti [6] proposed the Butterfly Optimization Algorithm (BOA) and Gases Brownian Motion Optimization for optimal threshold determination. This study validated the applicability of swarm intelligence algorithms in threshold optimization. Sharifi et al. [7] proposed the Cultural History Optimization Algorithm, a novel human-inspired metaheuristic featuring strong exploration and exploitation capabilities as well as rapid convergence. Abdel-Salam et al. [8] proposed the Elite-Adaptive-Turbulent Hiking Optimization Algorithm, which exhibited superior accuracy and efficiency, and the study investigated its application to multi-threshold intracerebral hemorrhage image segmentation. Currently, the swarm intelligence algorithms used in image segmentation have significant shortcomings in convergence rate and optimization efficiency. Therefore, it is necessary to improve and innovate algorithms to enhance their applicability.

The Slime Mould Algorithm (SMA), [9] inspired by the foraging behavior of slime molds, has a simple structure, few parameters, and high accuracy. It has been applied to image processing. [10] However, when it is applied to multi-peak and high-dimensional tasks, the algorithm shows shortcomings—sluggish convergence and a tendency to get stuck in local optima. To address these and boost SMA's optimization performance, Houssein et al. [11] proposed the Adaptive Guided Differential Evolution Algorithm. This integration resolved the imbalance between exploration and exploitation in the traditional SMA, significantly enhancing the performance of SMA. Guo et al. [12] proposed the Improved SMA, in which elite backward learning and quadratic interpolation were combined. The population diversity and local search capabilities of SMA were enhanced, and its global optimization ability was effectively improved. Naik et al. [13] developed an improved normalized square difference mechanism and a Leader SMA, proposing an efficient multilevel thresholding method that enhances segmentation accuracy while significantly reducing computational cost. Liu et al. [14] introduced a Modified Differential Evolution Algorithm inspired by the foraging behavior of the SMA, significantly improving the segmentation quality of breast cancer images. Hu et al. [15] developed the Dispersed Foraging SMA, demonstrating that the proposed methods achieved faster convergence, higher accuracy, and more compact feature subsets compared to traditional SMA. All the SMAs discussed previously are summarized in Table 1 for concise comparison. Although these studies optimized SMA using various strategies, SMA still faced insufficient convergence stability and a tendency to become trapped in local optima when dealing with complex multimodal problems. Its computational efficiency and accuracy in complex image segmentation still require further optimization.

Fractional-order calculus, an important branch of advanced mathematics, has wide-ranging applications in engineering, artificial intelligence, and medicine. Pu [16] first clarified the physical significance of fractional calculus in signal processing and its implementation algorithms, laying the foundation for its use in image processing. Li and Hou [17] applied fractional-order differentiation to image processing and optimized the structure of the Sobel operator, which laid a foundation for efficient and high-precision image edge detection. Wei et al. [18] put forward an adaptive order fractional differential strategy. It not only accelerated the convergence of the particle swarm algorithm but also improved optimization accuracy through a symmetric particle distribution mechanism. However, fractional-order calculus is highly effective at accelerating the convergence of swarm intelligence algorithms; its use in improving traditional algorithms remains limited. Few studies have combined it with intelligent algorithms for image segmentation. [19]

Strategy-based Fractional-order SMA (SFSMA) is introduced to overcome the traditional SMA's limitations in convergence speed and local optima entrapment. By integrating differential evolution strategies and fractional-order calculus into the SMA framework, both optimization accuracy and convergence rate are substantially improved. Subsequently, a high-precision SFSMA–Otsu segmentation model is developed through combining SFSMA with the two-dimensional (2-D) Otsu method, utilizing gray-gradient histograms to exploit image grayscale

Table 1. Comparative analysis of improved Slime Mould Algorithm variants

Literature	Algorithm	Improvement and problem-solving	Limitation
Chen et al. [10]	An improved Slime Mould Algorithm	Enhance global search capability and optimization accuracy	Decrease efficiency with increased threshold levels
Houssein et al. [11]	Adaptive Guided Differential Evolution Algorithm	Improve the local search capability and population diversity of the Slime Mould Algorithm	Difficult to adapt parameters dynamically to different optimization scenarios
Guo et al. [12]	Improved Slime Mould Algorithm	Enhance diversity, global optimization, and convergence accuracy; strengthen local exploitation, faster convergence, and local optima avoidance	Increase computational overhead; prone to excessive local exploitation in complex functions
Naik et al. [13]	Leader Slime Mould Algorithm	Reduce computation, improve accuracy, enhance exploitation capability, faster convergence	The stability of the algorithm relies on the initial population distribution, and its robustness still requires improvement
Liu et al. [14]	Modified Differential Evolution with Slime Mould Algorithm	Enhance convergence accuracy; improve the quality of breast cancer image segmentation	Prone to falling into local optima
Hu et al. [15]	Dispersed Foraging Slime Mould Algorithm	Improve convergence speed and accuracy, maintain population diversity; enhance classification accuracy, reduce selected feature count	Increases structural complexity, resulting in high sensitivity to parameter tuning

and spatial features. Comprehensive evaluation employs a two-stage experimental design: first, SFSMA is tested on 12 benchmark functions through 30 independent runs, with the best solution (BEST), the mean (MEAN), and the standard deviation (STD) quantifying optimization performance; second, the segmentation model is compared against six classical algorithms across multiple image types using peak signal-to-noise ratio (PSNR), mean squared error (MSE), structural similarity index measure (SSIM) and feature similarity index measure (FSIM) metrics. Results confirm SFSMA's superior convergence speed, optimization accuracy, and avoidance of local optima, establishing an effective solution for complex image segmentation.

The main contributions of this study are summarized as follows:

(i) A novel algorithm, SFSMA, was developed based on the SMA.
(ii) Through 30 independent repeated tests on 12 benchmark functions, SFSMA was thoroughly compared with several existing algorithms, confirming its high optimization accuracy and fast convergence.
(iii) Segmentation models were constructed by integrating SFSMA and other comparable algorithms with the 2-D Otsu algorithm. The superiority of SFSMA in convergence speed and segmentation accuracy was validated using four widely recognized evaluation metrics.
(iv) Significant improvements were achieved by SFSMA over the traditional SMA in both benchmark function optimization and image segmentation quality.

2. Two-dimensional Ootsu image segmentation

In practical image segmentation applications, the traditional one-dimensional Otsu algorithm relies solely on grayscale features and cannot effectively segment certain images. Wu et al. [20] replaced the original model with a joint histogram of grayscale levels and gradient values, thereby increasing the feature dimension. This enhanced the representation ability of edge features and significantly improved the segmentation accuracy.

For a digital image of size $M \times N$ with L grayscale levels, the coordinates (i, j) correspond to the joint probability density p_{ij} Equation (1):

$$p_{ij} = \frac{n_{ij}}{M \times N} \tag{1}$$

The binary pair of grayscale threshold(s) and gradient threshold (t) splits the image into two regions: target object and background. After simplifying the computational complexity, the joint probability expressions for the background class probability P_A and the target class probability P_B are shown in Equation (2)).

$$P_A(s,t) = \sum_{i=1}^{s}\sum_{j=1}^{t} p_{ij}, P_B(s,t) = \sum_{i=s+1}^{L}\sum_{j=t+1}^{L} p_{ij} \tag{2}$$

The mean vectors of μ_A and μ_B are calculated as shown in Equations (3) and (4), respectively

$$\mu_A(s,t) = (\mu_{Ai}, \mu_{Aj})^T = \left(\sum_{i=1}^{s}\sum_{j=1}^{t} \frac{ip_{ij}}{P_A}, \sum_{i=1}^{s}\sum_{j=1}^{t} \frac{jp_{ij}}{P_A} \right)^T \tag{3}$$

$$\mu_B(s,t) = (\mu_{Bi}, \mu_{Bj})^T = \left(\sum_{i=1}^{s} \sum_{j=1}^{t} \frac{ip_{ij}}{P_B}, \sum_{i=1}^{s} \sum_{j=1}^{t} \frac{jp_{ij}}{P_B} \right)^T \tag{4}$$

The total mean vector of $(s,\ t)$ is given by Equation (5).

$$\mu(s,t) = (\mu_i, \mu_j)^T = \left(\sum_{i=1}^{L} \sum_{j=1}^{L} ip_{ij}, \sum_{i=1}^{L} \sum_{j=1}^{L} jp_{ij} \right)^T \tag{5}$$

The inter-class dissimilarity matrix describing the distinction between the background and target classes is provided in Equation (6).

$$\begin{aligned} S(s,t) &= P_A \times (\mu_A - \mu) \times (\mu_A - \mu)^T \\ &+ P_B \times (\mu_B - \mu) \times (\mu_B - \mu)^T \end{aligned} \tag{6}$$

The trace of the dissimilarity matrix is shown in Equation (7).

$$\begin{aligned} \mathrm{tr}(S(s,t)) &= P_A \left[(\mu_{Ai} - \mu_i)^2 + (\mu_{Aj} - \mu_j)^2 \right] \\ &+ P_B \left[(\mu_{Bi} - \mu_i)^2 + (\mu_{Bj} - \mu_j)^2 \right] \end{aligned} \tag{7}$$

The optimal segmentation threshold is determined by maximizing the trace, as shown in Equation (8).

$$tr\left(S\left(s^*, t^*\right)\right) = \max\left(tr\left(S\left(s,t\right)\right)\right) \tag{8}$$

3. Strategy-based fractional-order slime mould algorithm

3.1. Slime Mould Algorithm

The SMA constructs a swarm-intelligence optimization model by simulating the foraging behavior of the slime mold and the dynamic characteristics of the biological oscillator that controls cytoplasmic flow. The mechanism of this algorithm is shown through three levels of dynamic coordination. In the phase of approaching food, the individual's position is determined by the global optimal solution $x_b\left(t\right)$ a and two randomly selected individuals $x_A\left(t\right)$ and $x_B\left(t\right)$, as shown in Equation (9):

$$x(1+t) = \begin{cases} x_b(t) + v_b(t) \times \left(W \times X_A(t) - X_B(t)\right), & r < p \\ v_c \times x(t), & r \geq p \end{cases} \tag{9}$$

where:
v_b is a random number $[-a, a]$ within the range;
W is the weight coefficient of the slime mold;
v_c is a parameter that decreases linearly from 1 to 0;
r is a random number between 0 and 1.

The probability threshold p is then calculated by Equation (10):

$$p = \tanh\left|S(i) - F_D\right| \tag{10}$$

where:
$S\left(i\right)$ is the fitness value of the $i = (1, 2, ...N)$ slime mold individual;
F_D is the best fitness of the slime mold in all previous iterations.

The updated formula for the value of the parameter a is shown in Equation (11):

$$a = \arctan h\left(-(\frac{t}{T_{\max}}) + 1\right) \tag{11}$$

The a decays nonlinearly with the number of iterations t. The algorithm maintains strong perturbations in the early stages to explore a wide solution space, and then gradually carries out local refinement searches in the later stages. W is calculated using Equations (3) and (4), the dynamic relationship between the individual fitness $S\left(i\right)$ and the population extremes bF and wF is converted into adaptive search weights W, which enhances the development capabilities of high-fitness regions Equation (12).

$$W\left(SI(i)\right) = \begin{cases} 1 + r \times log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & i = C \\ 1 - r \times log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & i = 0 \end{cases} \tag{12}$$

In the food encirclement stage, the introduction factor ($z = 0.03$) balances global search and local search. The formula for updating the slime mold's position at this stage is as shown in Equation (13):

$$x\left(t+1\right) = \begin{cases} rand\left(B_U - B_L\right) + B_U, & r < z \\ x_b\left(t\right) + v_b\left(W \cdot x_A\left(t\right) - x_B\left(t\right)\right), & rand \geq z, r < p \\ v_c \cdot x\left(t\right), & rand \geq z, r \geq p \end{cases} \tag{13}$$

The cooperative search strategy of the slime mold vein network avoids falling into a local optimum while keeping global search potential by dynamically adjusting population diversity.

3.2. Fractional-order calculus

Fractional-order calculus goes beyond the dimensional limits of traditional integer calculus, extending the order of differentiation and integration of functions. Its definition forms differ based on the mathematical context and practical needs. The Grünwald–Letnikov (G—L) type is based on extending the limit of integer-order difference approximations, using the positivity or negativity of the order parameter $a \in R$ to represent fractional

differentiation and fractional integration, respectively. Its discretization features offer computational advantages in areas such as digital signal processing. The Caputo type solves the initial-value adaptability problem by first performing fractional integration of integer-order derivatives, thereby making the initial-value conditions of the differential equation compatible with traditional theory and becoming a core tool for engineering dynamics modeling. Among the three definitions, the numerical efficiency of the G–L type and the physical interpretability of the Caputo type serve control engineering and system modeling, respectively, while the Riemann–Liouville type provides a mathematical basis for theoretical analysis. These three types complement each other and meet the multiscale modeling needs of complex systems.

Ren et al. [21] proposed that the optimization performance of the SMA fundamentally relies on its capability of dynamic adaptation between global exploration and local exploitation. The optimization performance of SMA was enhanced by them through the integration of Gaussian Kernel probability strategies and novel motion mechanisms, providing a critical theoretical reference for selecting fractional-order calculus types in this study. In the context of SMA optimization, the G–L fractional-order operator expands the search range by adjusting fractional-order parameters, thereby reducing the risk of the algorithm becoming trapped in local optima. During the local exploitation phase, the Caputo-type fractional-order operator exhibits heightened sensitivity to local parameter changes, enabling fine-grained convergence control to ensure optimization accuracy. In contrast, the Riemann–Liouville (R–L) definition of fractional calculus serves as a theoretical cornerstone. Its initial conditions depend on abstract fractional-integral terms, and its computation necessitates the introduction of complex integral kernels. Therefore, the G–L and Caputo fractional-order calculi were selected in this study to optimize the motion strategy of SMA, as their respective characteristics synergistically address the core requirement of balancing global exploration and local exploitation.

3.2.1. Grünwald–Letnikov-type definition

The G–L definition is based on the concept of integer-order differentiation. In a previous study by Cao et al., [22] the m-order derivative of the function $f(t)$ was defined by the G–L fractional order as Equation (14):

$$\frac{d^m f(t)}{dt^m} \equiv \lim_{h\to 0}\left\{ h^{-m}\sum_{j=0}^{\infty}(-1)^j \binom{m}{j} f(t-jh)\right\} \tag{14}$$

where:
m is of any order;
h is the step size;
$\binom{m}{j}$ is the binomial coefficient.

Expanding the binomial results in $\binom{m}{j} = \frac{m(m-1)\cdots(m-j+1)}{j!}$, where $j \geq 1$. Replacing the integer order m with the fractional order v and using the properties of the Gamma function, Equation (14) can be simplified to Equation (15):

$$\begin{aligned} {}_a^G D_t^v f(t) = \lim_{h\to 0} h^{-v} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \\ \frac{\Gamma(v+1)}{\Gamma(j+1)\Gamma(v-j+1)} f(t-jh) \end{aligned} \tag{15}$$

when $h=1$, let $n=t-a$, the fractional order derivative dispersion expression of the function is Equation (16).

$$D^v[f(t)] = h^{-v}\sum_{j=0}^{n}\frac{(-1)^j\Gamma(v+1)}{j!\Gamma(v-j+1)} f(t-jh) \tag{16}$$

3.2.2. Caputo-type definition

Caputo's definition is an alternative to the G–L definition. It simplifies the form of the Laplace transform, thereby simplifying the solution process for fractional-order differential equations, improving its applicability in modern engineering. The Caputo type first performs integer-order differentiation, then fractional-order integration. The definition by Chen et al. [23] is as follows Equation (17):

$$\begin{aligned} &{}_a D_t^v f(t) = \frac{1}{\Gamma(m-v)}\int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{v-m+1}} d\tau, \\ &(m-1) \leq v < m \end{aligned} \tag{17}$$

Let $m=1$, Equation (17) can then be rewritten as Equation (18).

$${}_a D_t^v f(t) = \frac{1}{\Gamma(1-v)}\int_0^t \frac{1}{(t-\tau)^v} f'(\tau)\, d\tau \tag{18}$$

Divide interval $[0,t]$ into N equal parts, each part being $\Delta t = \frac{t}{N}$; when $\Delta t = 1$, $t_k = k\frac{t}{N} = k$,

the $N+1$ equal division nodes $\{f_0, f_1 \ldots f_N\}$. Using the forward-difference format of the first-order derivative, Equation (19) can be obtained.

$$\int_{kx/N}^{(kx+1)/N} \frac{f(\tau)}{(t-\tau)^{v-2}} d\tau \approx \frac{f(t-t_{k+1}) - f(t-t_k)}{(1-v)(\Delta t)^v} \left[(k+1)^{1-v} - k^{1-v}\right] \tag{19}$$

Substituting Equation (19) into Equation (18), an approximate form of the Caputo-type fractional-order differential of the function $f(t)$ can be derived. The fractional-order differential operation can then be simplified to addition and multiplication operations, which are easier to implement on a computer, as shown in Equation (20).

$${}_aD_t^v f(t) \approx \sum_{k=0}^{N-1} c_k f_k \approx f(t) \times C \tag{20}$$

Approximate the fractional order as the convolution of $f(t)$ and C, the coefficient function C can be calculated as follows Equation (21):

$$\begin{cases} c_0 = \dfrac{-1}{\Gamma(2-v)} \\ c_1 = \dfrac{2-2^{1-v}}{\Gamma(2-v)} \\ \vdots \\ c_k = \dfrac{2(N-k)^{1-v} - (N-k+1)^{1-v} - (N-k-1)^{1-v}}{\Gamma(2-v)} \\ \vdots \\ c_N = \dfrac{N^{1-v} - (N-1)^{1-v}}{\Gamma(2-v)} \end{cases} \tag{21}$$

3.2.3. Differential evolution

From the perspective of the SMA mechanism, in the early stage, under the influence of the contraction factor z, slime mold individuals focus on global search; in the later stage, as z decays, they focus on local search. The global search capability in the early stage is relatively weak, failing to fully explore the solution space, and in the later stage, it tends to become trapped in local optima, which in turn results in poor population diversity. To address these shortcomings, this paper proposes to reconstruct the SMA position update method using the DE/current/to/best1 differential evolution strategy, calculating the population difference vector to generate perturbations. The advantage of this approach is that it ensures the stability of local search, while the random component effectively prevents premature convergence of the population through perturbations. The updated method is shown in Equation (22):

$$\begin{aligned} v_i(t+1) &= x_i(t) + F \times (x_{best} - x_i(t)) \\ &+ F \times (x_{r1}(t) - x_{r2}(t)) \ (22) \end{aligned} \tag{22}$$

where x_{r1} and x_{r2} are random slime molds. The step size F can control the range and search accuracy of the algorithm, as shown in Equation (23).

$$F = 0.9 + (0.9 - 0.5) \times (1 - \frac{t}{T_{\max}}) \tag{23}$$

where $T_{\max}$ represents the maximum iteration count, t represents the current iteration count, and F with a value range of $[0.5, 0.9)$. The step size is linearly reduced from the initial value to 0.5 as the iteration process progresses, thus enhancing the global search capability in the early stages of the algorithm through a larger step size. As the iteration progresses, the step size is gradually reduced to improve local search accuracy, avoiding the problems of premature convergence or search stagnation caused by a fixed step size. The optimized position update in Equation (24) introduces a greedy mechanism, as shown below:

$$x_i(t+1) = \begin{cases} x_i(t+1), f(x_i(t+1)) > f(v_i(t+1),) \\ v_i(t+1), f(x_i(t+1)) \le f(v_i(t+1),) \end{cases} \tag{24}$$

Equation (24) uses the current target individual as the reference vector and generates a perturbation term by introducing deterministic differential vectors and stochastic differential vectors. This optimizes global search capabilities and local search efficiency.

3.3. Development of Strategy-based fractional-order slime mould algorithm

The traditional SMA primarily relies on current food concentration gradients and random oscillations for position updates during local search, with historical search states not fully utilized. This can lead to slowed convergence in later stages, low local search efficiency, and entrapment in local optima. Three key improvements were made to the SMA in this study: First, differential evolution and G–L-type fractional-order calculus are combined to modify the SMA's position update method, thereby enhancing its convergence speed; Second, Caputo-type fractional-order calculus is applied to improve the SMA's position update approach, strengthening the local optimization capability of slime mold individuals; Third, to prevent the algorithm from being trapped in

local optimal solutions, a dynamic threshold adjustment mechanism is introduced into the framework.

3.3.1. G-L type fractional-order optimization

By introducing fractional calculus, an individual's search is not only determined by its current state but also influenced by its past effective paths. This approach enhances the directionality of the search by incorporating memory of historical states, significantly improving the algorithm's convergence speed and ability to escape local optima in complex problems, thus optimizing overall optimization performance.

As indicated by Equation (13), the formula used to update the local search position ($r < p$) of the SMA can be expressed through Equation (25).

$$\begin{aligned} x_i(t+1) - x_i(t) &= F \times (x_{best} - x_i(t)) \\ &+ F \times (x_{r1}(t) - x_{r2}(t)) \end{aligned} \tag{25}$$

The expression in Equation (25) is treated as the first-order difference form $D^n\left[x\left(t+1\right)\right]$ within the G–L fractional calculus framework. The expression can be further simplified using Equation (16). When n is set to 4, the expression $D^n\left[x\left(t+1\right)\right]$ can be equivalently expressed as Equation (26).

$$\begin{aligned} &D^v\left[x(t+1)\right] \approx x(1+t) - vx(t) \\ &+ \frac{1}{2}v(v-1)x(t-1) - \frac{1}{6}v(v-1)(v-2)x(t-2) \\ &+ \frac{1}{24}v(v-1)(v-2)(v-3)x(t-3) \end{aligned} \tag{26}$$

Equations (25) and (26) can be used to derive the optimized SMA's local position update formula Equation (27):

$$\begin{aligned} x_i(t+1) &= vx_i(t) - \frac{1}{2}v(v-1)x_i(t-1) \\ &+ \frac{1}{6}v(v-1)(v-2)x_i(t-2) \\ &- \frac{1}{24}v(v-1)(v-2)(v-3)x_i(t-3) \\ &+ F \times (x_{best} - x_i(t)) + F \times (x_{r1}(t) - x_{r2}(t)) \end{aligned} \tag{27}$$

As shown from Equation (27), the fractional order has an impact on the position update of slime mold individuals. To obtain better results, an evolution factor is introduced. This factor adjusts the fractional order adaptively based on the position information of slime mold individuals. The average distance between slime mold individual i and other slime mold individuals is shown in Equation (28):

$$d_{ix} = \frac{1}{N-1} \sum_{j=1, j \equiv i}^{N} \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2} \tag{28}$$

Among them, N is the number of slime mold individuals, D is the spatial dimension. The evolutionary factor f_y is shown in Equation (29).

$$f_y = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \tag{29}$$

where d_g is defined as the average distance from the globally optimal slime mold's position to all other individuals in the group. For set d_{ix}, $d_{\max}$, and $d_{\min}$ are assigned to their maximum and minimum values, respectively. When the order$v \in [0.5, 0.8]$ takes the value, the convergence speed achieves its maximum. The calculation and derivation of the adaptive order are formally illustrated in Equation (30).

$$v(f_y) = 0.5 + 0.3\cos\left(\frac{\pi}{2}\left(1 - \exp\left(-0.47 f_y\right)\right)\right) \tag{30}$$

Minor variations in the parameter are amplified by the evolutionary factor, thereby exerting a pronounced influence on position updates. When $v(f_y)$ is slightly increased, the exploratory behavior of position updates is enhanced, though convergence speed may be reduced due to excessive divergence. Conversely, a slight decrease in $v(f_y)$ prioritizes convergence behavior but may lead to premature convergence to local optima due to insufficient exploration. Compared with fixed-order approaches, this adaptive mechanism based on specific operating points ensures that model output remains both stable and optimal, even under minor parametric perturbations.

3.3.2. Caputo-type fractional-order optimization local search

The local search diversity of individuals is enhanced by Equation (13) when $r \geq \rho$. However, this formulation does not adjust the perturbation amplitude. This leads to inadequate search in the early phase and unstable convergence in the later stage. The Caputo-type definition, which features memory effects and nonlinear characteristics, is considered a refinement of the G–L-type definition. In the present study, the Caputo-type definition is incorporated into the iterative formula of the SMA. The perturbation amplitude is dynamically adjusted based on the iterative stage or population diversity, thereby effectively improving the algorithm's optimization performance. According to Equation (21), the difference form of the Caputo-type fractional derivative of the function when $N = 3$ is given by

Equation (31), and its difference form is shown in Equation (32).

$$D^{v}[x(t)] = -c_0 x(t) + c_1 x(t-1) + c_2 x(t-2) \tag{31}$$

$$D^{v}[x(t+1)] = -c_0 x(t+1) + c_1 x(t) + c_2 x(t-1) \tag{32}$$

According to the $r \geq p$ individual position update method of the SMA, the deformation form shown in Equation (33) is obtained

$$x\left(t+1\right) - x\left(t\right) = \left(1 - v_c\right) x\left(t\right) \tag{33}$$

From Equations (31) and (32), the improved Caputo-type fractional-order position update formula is derived; this is the fractional-order SMA's local update method, with its formula shown in Equation (34):

$$x\left(1+t\right) = \frac{\left(1 + c_1 - v_c\right) x\left(t\right) + c_2 x\left(t-1\right)}{c_0} \tag{34}$$

where $c_0 = -1/\Gamma\left(2-v\right)$;
$c_1 = \left(2 - 2^{1-v}\right)/\Gamma\left(2-v\right)$;
$c_2 = \left(2 \times 2^{1-v} - 3^{1-v} - 1\right)/\Gamma\left(2-v\right)$.

3.3.3. Dynamic threshold-adjusted local optimum strategy

During iterative optimization of the SMA, the rate of increase in fitness values often exhibits a phased decay, with the search range continuously contracting. To address the above shortcomings, a strategy to escape local optima through dynamic threshold adjustment is proposed. This is achieved by introducing a threshold observation model based on iterative difference sequences for dynamic monitoring. When the number of iterations reaches four, the detailed steps for this process are presented as follows Equations (35)–(37):

Step 1: Let $\Delta f\left(t\right)$ represent the fitness difference between the algorithm's t-th iteration and its $(t-3)$-th iteration.

$$\Delta f\left(t\right) = \left| f\left[x\left(t\right)\right] - f\left[x\left(t-3\right)\right] \right| \tag{35}$$

Step 2: Set $\theta\left(t\right)$ as the fitness value monitoring factor.

$$\theta\left(\mathrm{t}\right) = 0.05 \exp\left[-8\left(t-3\right)/T_{\max}\right] \tag{36}$$

Step 3: Make a comparison between the magnitudes of $\Delta f(t)$ and $\theta\left(t\right)$. If $\Delta f(t)$ ¡ $\theta\left(t\right)$, the algorithm is considered to have fallen into a local optimum, and the position update of the individual is perturbed using the fractional order in Equation (26). The position update formula is shown in Equation (37).

$$\begin{aligned} x_i\left(t+1\right) = {} & v x_i\left(t\right) - \frac{1}{2} v\left(v-1\right) x_i\left(t-1\right) \\ & + \frac{1}{6} v\left(v-1\right)\left(v-2\right) x_i\left(t-2\right) \\ & - \frac{1}{24} v\left(v-1\right)\left(v-2\right)\left(v-3\right) x_i\left(t-3\right) \end{aligned} \tag{37}$$

This strategy identifies if the algorithm has fallen into a local optimum and introduces fractional differential perturbations to solutions. By adopting this approach, the diversity within the population can be effectively increased. When $\theta\left(t\right)$ is slightly increased, the local escape mechanism is triggered earlier, which enhances the algorithm's ability to escape from local optima, though potentially at the cost of reduced search stability due to oversensitivity. Conversely, when $\theta\left(t\right)$ is slightly decreased, the identification of local optima is delayed. While this helps maintain smoother search progression, it increases the risk of becoming trapped in suboptimal regions. Consequently, the risk that the algorithm will get stuck in a local optimal solution is substantially lowered.

3.4. Detailed steps of the Strategy-based fractional-order slime mould algorithm

To address the slow convergence and susceptibility to local optima in the SMA, the SFSMA is proposed. In the SFSMA, a differential evolution strategy is incorporated to enhance population diversity, while fractional-order calculus is introduced to improve the optimization capability of the population. A dynamic threshold is utilized to monitor the algorithm's state and prevent entrapment in local optima. The detailed implementation steps of the SFSMA are outlined as follows:

Step 1: Initialize the slime mold population and associated parameters, specifying a population size of 30 and a maximum iteration count of 100.
Step 2: Calculate the fitness function value for each slime mold individual and sort these values. Next, set the position of the slime mold individual with the best fitness value as the current global optimal position, and take its fitness value as the destination fitness.
Step 3: After each iteration, update the optimal position of the slime mold individual. If $r < z$, conduct Equation (13).

Step 4: When $r < p$, introduce the position update Equations (27) and (13). Then select the offspring with the highest fitness based on the greedy principle.
Step 5: When $r \geq p$, use the position update Equation (34) that incorporates fractional order for local search. Meanwhile, cache the optimal individual position and fitness value of each iteration.
Step 6: When the number of iterations exceeds four, a dynamic threshold monitoring strategy is adopted. If it is identified that the algorithm has fallen into a local optimum, perform individual perturbation using Equation (37).
Step 7: If the current iteration count is lower than the maximum iteration count, Steps 3–7 should be executed. Once the iteration count reaches the maximum value, the loop is terminated, and the global optimal position and fitness value are output.

The flow chart for the SFSMA–Otsu is presented in Figure 1.

3.5. Time complexity analysis

Based on the slime mold population size N, problem dimension D, and maximum iteration T, the time complexity of the proposed SFSMA is analyzed as follows. The initialization of population positions is characterized by $O(N)$ time complexity. The evaluation of fitness values and the subsequent sorting operation require $O(N + \log N)$ time complexity. The calculation of weight coefficients for all individuals demands $O(N \times D)$ time complexity. The computation of adaptive fractional orders is achieved with $O(N)$ time complexity. The position update process is completed with $O(N \times D)$time complexity. The generation of offspring through differential evolution maintains $O(N)$ time complexity. The stagnation detection using a dynamic threshold mechanism requires $O(1)$ time complexity. In summary, the overall time complexity of the SFSMA per iteration is Equation (38).

$$O(D + T(3N + N\log N + 2ND)) \tag{38}$$

while both the SMA and the SFSMA are maintained within the same computational complexity order.

4. Evaluation of improved algorithms for benchmark functions

In this paper, benchmark functions were used to verify the optimization performance of the fractional-order optimized SFSMA. The single-mode benchmark functions (F1–F4) were applied to evaluate the convergence efficiency of the algorithm, while the multi-mode benchmark functions (F5–F12) were used to test its global search ability and solution accuracy. The experiments were carried out in a comparative framework, where the SFSMA was compared to the traditional SMA, standard particle swarm optimization (PSO) algorithm, FA, Fractional FA (FFA), and the HFBOA presented by Ma et al. [4] To mitigate experimental randomness, all involved algorithms were compared under identical conditions: the population size was set to 30, the maximum number of iterations was uniformly fixed at 500, and each algorithm was independently tested 30 times on the benchmark functions. Detailed parameters are presented in Table 2.

Three indices were recorded: the BEST, MEAN, and STD. These indices were used to quantitatively evaluate the optimization accuracy and robustness of the algorithms. In addition, the convergence characteristics of the algorithms were analyzed through fitness curves. The performance data of all algorithms on the benchmark function set are given in Table 3.

Based on the experimental results summarized in Table 3 and the convergence behavior illustrated in Figure 2, the performance of all algorithms on unimodal functions F1–F4 was evaluated. Both SFSMA and HFBOA achieved the theoretical optimum value of 0, with optimal MEAN and STD values, indicating high solution accuracy and stability. However, in terms of convergence speed, SFSMA demonstrated superior efficiency, requiring fewer iterations to reach the optimum compared to HFBOA. Compared to conventional algorithms, such as SMA and PSO, the advantage of SFSMA was more pronounced. Although SMA attained relatively high precision, it consistently failed to reach the theoretical optimum. In contrast, SFSMA maintained rapid convergence while consistently achieving the theoretical optimum, underscoring its overall effectiveness in unimodal function optimization.

In tests on the multimodal functions F5–F8, the algorithms showed significant performance divergence. On F6 and F7, both SFSMA and HFBOA achieved the theoretical optimum, showing similar performance. On F7 and F8, SFSMA and SMA achieved identical core metrics and also attained the theoretical optimum. However, SFSMA demonstrated significantly faster convergence, requiring markedly fewer iterations.

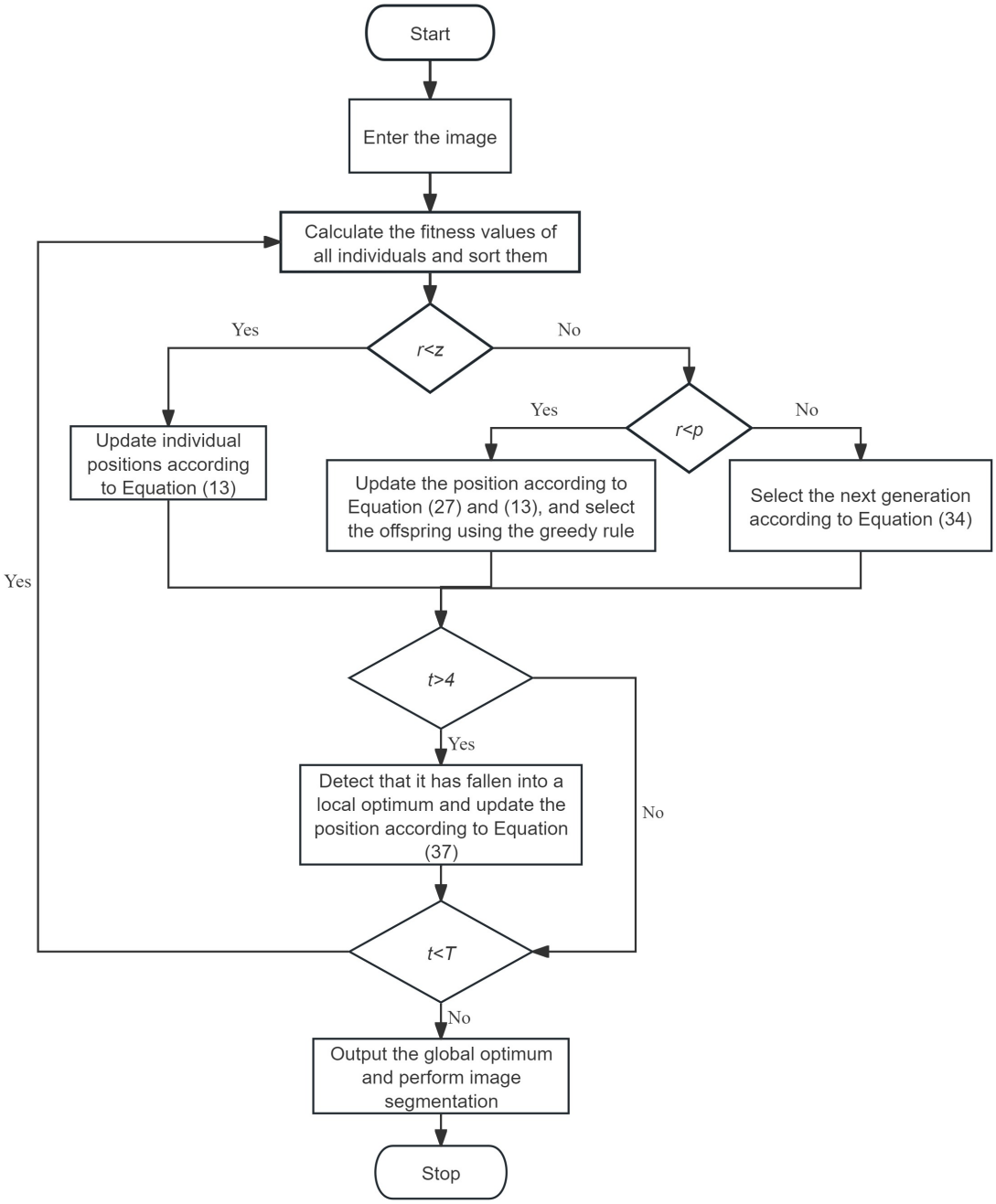

Figure 1. The flow chart for the Strategy-based Fractional-order Slime Mould Algorithm–Otsu

The advantage of SFSMA was particularly pronounced on F5 and F8, where both its BEST and MEAN values consistently achieved the theoretical optimum, and its STD remained the lowest among all compared algorithms. In contrast, the MEAN values of HFBOA fell into a non-optimal range. Several algorithms, such as PSO, FA, and FFA, failed to achieve the theoretical optimum in both BEST and MEAN, largely due to their susceptibility to local optima.

These performance differences could be attributed to the alignment between the algorithm's mechanisms and its function characteristics. Although F6–F8 were multimodal, they contained few local optima and exhibited smooth, sparse landscapes. Under such conditions, the fractional-order mechanisms in HFBOA and the core design of SMA were well-suited to deliver high accuracy. However, SMA relied on passive information exchange among individuals, leading to redundant exploration in early stages due to large step sizes, whereas the decay of slow step sizes in later stages prolongs

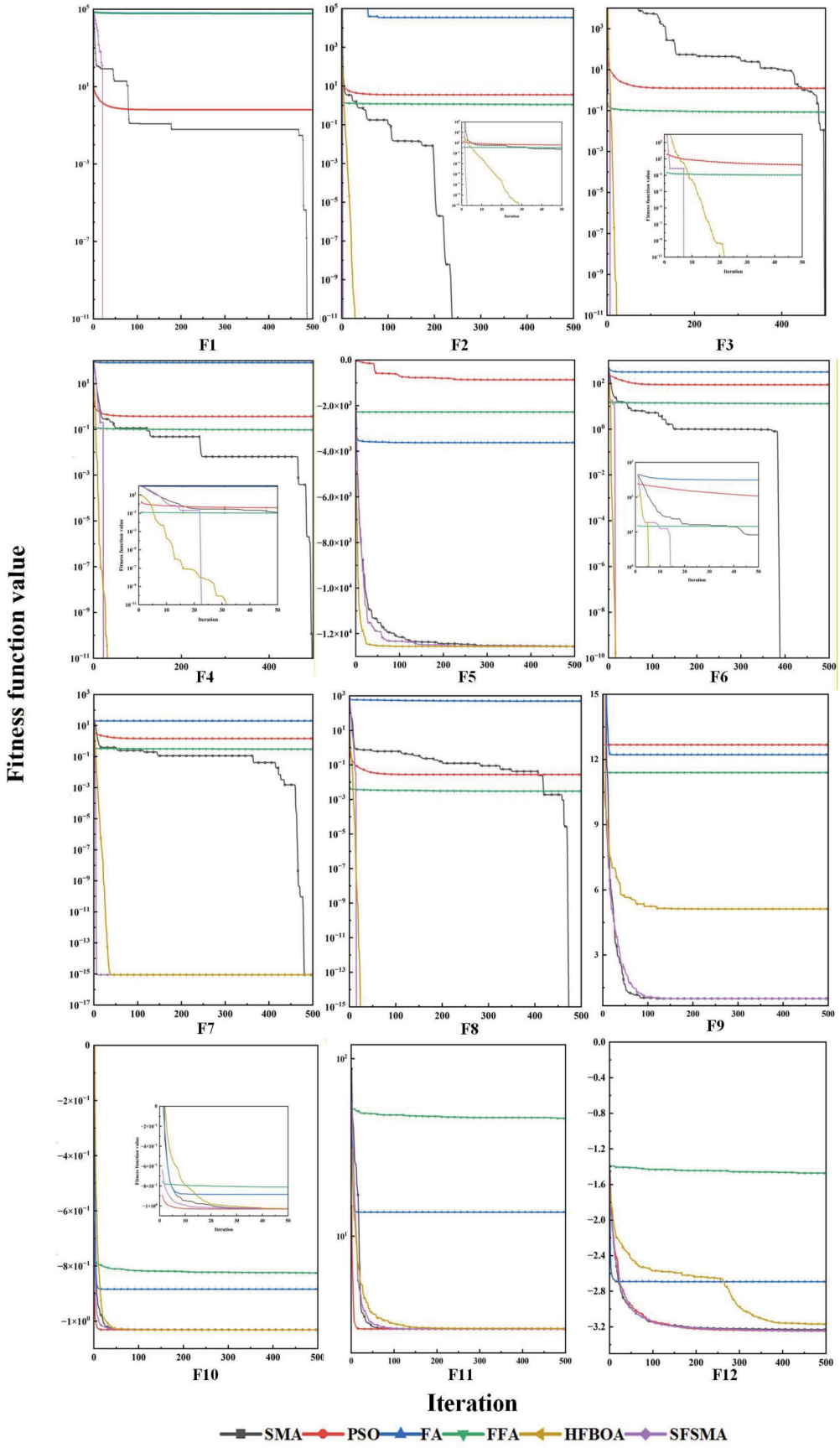

Figure 2. Convergence curves of each algorithm in benchmark functions F1–F12
Abbreviations: FA, Firefly Algorithm; HFBOA, Hybrid Fractional-order Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm.

Table 2. Basic information on benchmark functions

Serial number	Function	Dimension	Range	f_{min}
F1	Sphere function	30	[−100, 100]	0
F2	Schwefel's problem 2.22	30	[−10, 10]	0
F3	Schwefel's problem 1.2	30	[−100, 100]	0
F4	Schwefel's problem 2.21	30	[−100, 100]	0
F5	Generalized Schwefel's problem 2.26	30	[−500, 500]	−12,569.5
F6	Generalized Rastrigin's function	30	[−5.12, 5.12]	0
F7	Ackley's function	30	[−32, 32]	0
F8	Generalized penalized function 1	30	[−50,50]	0
F9	Shekel's foxholes function	2	[−65, 65]	1
F10	Six-hump camel-back function	2	[−5, 5]	−1.0316
F11	Goldstein-price function	2	[−2, 2]	3
F12	Hartmann function	6	[0, 1]	−3.3224

Table 3. Comparison of the benchmark function results

Function	Index	SMA	PSO	FA	FFA	HFBOA	SFSMA
	BEST	1.2008×10^{-291}	0.1542	3.2832×10^{4}	0.0629	0	0
F1	MEAN	5.2986×10^{-60}	0.7459	5.9570×10^{4}	0.0664	0	0
	STD	2.0781×10^{-59}	0.3551	8.6746×10^{3}	0.0077	0	0
	BEST	2.3561×10^{-300}	2.5909	84.0761	0.9446	0	0
F2	MEAN	1.0625×10^{-143}	3.8591	3.4100×10^{4}	1.1173	0	0
	STD	5.8196×10^{-143}	0.5787	1.6895×10^{5}	0.0618	0	0
	BEST	9.3498×10^{-192}	0.4383	4.8350×10^{4}	0.0582	0	0
F3	MEAN	2.3336×10^{-23}	1.1249	1.0811×10^{5}	0.0843	0	0
	STD	1.1938×10^{-22}	0.4349	3.0665×10^{4}	0.0100	0	0
	BEST	1.6598×10^{-111}	0.2572	65.8019	0.0863	0	0
F4	MEAN	8.4435×10^{-19}	0.3999	81.3071	0.0958	0	0
	STD	4.5976×10^{-18}	0.0785	5.3138	0.0039	0	0
	BEST	-1.2569×10^{4}	-1.6985×10^{4}	-4.9963×10^{3}	-3.3864×10^{3}	-1.2569×10^{4}	-1.2569×10^{4}
F5	MEAN	-1.2565×10^{4}	-8.6421×10^{2}	-3.6240×10^{3}	-2.2753×10^{3}	-1.2566×10^{4}	-1.2567×10^{4}
	STD	7.2443	3.1222×10^{3}	6.0045×10^{2}	5.4530×10^{2}	7.01727	6.2076
	BEST	0	53.3354	275.9369	10.5653	0	0
F6	MEAN	8.6023×10^{-13}	89.7438	326.6720	12.9767	0	0
	STD	4.7117×10^{-12}	17.6245	6.1096×10^{7}	0.7162	0	0
	BEST	8.8816×10^{-16}	0.5123	18.9686	0.2687	8.8816×10^{-16}	8.8816×10^{-16}
F7	MEAN	8.8816×10^{-16}	1.4448	19.9743	0.3012	8.8816×10^{-16}	8.8816×10^{-16}
	STD	0	0.3557	0.1923	0.0207	0	0
	BEST	0	0.0137	336.8269	0.0022	0.0095	0
F8	MEAN	0	0.0327	520.0291	0.0030	0.0268	0
	STD	0	0.0126	87.6597	2.8728×10^{-4}	0.0121	0
	BEST	0.9980	12.6705	0.9980	3.9691	0.9980	0.9980
F9	MEAN	0.9980	12.6705	12.2195	11.3999	4.6339	0.9980
	STD	3.5149×10^{-12}	1.5175×10^{-13}	8.4558	2.7773	4.3109	2.7178×10^{-12}
	BEST	−1.0316	−1.0316	−0.9923	−0.8974	−1.0316	−1.0316
F10	MEAN	−1.0316	−1.0316	−0.8254	−0.8065	−1.0315	−1.0316
	STD	5.2378×10^{-8}	6.6496×10^{-16}	0.0551	0.0354	2.1630×10^{-4}	1.0480×10^{-9}
	BEST	3	3	3.0001	3.3161	3.0001	3
F11	MEAN	3	3	13.6625	46.1763	3.0171	3
	STD	2.5037×10^{-9}	1.2960×10^{-15}	26.5722	32.9740	0.0257	1.7017×10^{-9}
	BEST	−3.3220	−3.3220	−3.2203	−2.7884	−3.3127	−3.3220
F12	MEAN	−3.2345	−3.2426	−2.6928	−1.4718	−3.1697	−3.2426
	STD	0.0555	0.0605	0.2582	0.4652	0.0823	0.0537

Abbreviations: BEST, best solution; FA, Firefly Algorithm; FFA, Fractional Firefly Algorithm;
HFBOA, Hybrid Fractional-order Butterfly Optimization Algorithm; MEAN, mean;
PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm;
SMA, Slime Mould Algorithm; STD, standard deviation.

convergence. In comparison, SFSMA retained SMA's core strengths while incorporating dynamic weight adjustment and a fractional-order memory operator. The dynamic weight mechanism adaptively reduced step size to minimize exploration redundancy, and the fractional-order memory leveraged historical information to guide convergence. Consequently, SFSMA matched the optimization accuracy of SMA and HFBOA while converging significantly faster, demonstrating enhanced suitability for complex optimization scenarios.

Table 4. Objective evaluation indices of human image segmentation results

Image	Algorithm	Iterations	Fitness	PSNR	MSE	SSIM	FSIM
Lena	SMA–Otsu	42	2622.4352	8.9973	8191.3424	0.1130	0.6576
	PSO–Otsu	97	2622.2150	8.9949	8195.8089	0.1110	0.6553
	BOA–Otsu	36	2622.2150	8.9949	8195.8089	0.1110	0.6553
	WOA–Otsu	75	2621.7157	8.9979	8190.1153	0.1145	0.6504
	Im-Fpso–Otsu	11	2620.1344	8.9976	8190.5955	0.1124	0.6574
	FFA–Otsu	2	2612.6587	8.9236	8331.5129	0.0941	0.6378
	SFSMA–Otsu	5	2622.4352	9.0129	8161.8818	0.1144	0.6587
Blonde-woman	SMA–Otsu	24	2522.3062	8.8670	8440.7341	0.0840	0.6042
	PSO–Otsu	82	2522.2360	8.8703	8434.2968	0.0822	0.6017
	BOA–Otsu	3	2517.7218	8.8138	8544.6917	0.0672	0.5937
	WOA–Otsu	2	2509.3020	8.8519	8470.1980	0.0964	0.6007
	Im-Fpso–Otsu	76	2521.8926	8.8708	8433.3908	0.0812	0.5994
	FFA–Otsu	11	2521.8926	8.8530	8468.0008	0.0752	0.5947
	SFSMA–Otsu	5	2522.3062	8.8759	8423.5147	0.0833	0.6052
Pirate	SMA–Otsu	69	4939.6046	8.4858	9215.1555	0.0756	0.6055
	PSO–Otsu	87	4938.7972	8.4436	9305.0194	0.0732	0.6024
	BOA–Otsu	44	4933.3418	8.4454	9301.2698	0.0643	0.5866
	WOA–Otsu	2	4940.1107	8.5122	9159.3206	0.0850	0.6204
	Im-Fpso–Otsu	36	4938.9832	8.4892	9207.9594	0.0736	0.6015
	FFA–Otsu	2	4930.7275	8.4610	9267.9058	0.0747	0.6041
	SFSMA–Otsu	13	4941.2732	8.5298	9122.2525	0.0761	0.6056
Couple	SMA–Otsu	59	1227.4382	8.3340	9542.9877	0.0785	0.6063
	PSO–Otsu	52	1226.6058	8.4552	9280.2408	0.0755	0.6060
	BOA–Otsu	68	1226.7321	8.4035	9391.4073	0.0639	0.6034
	WOA–Otsu	38	1227.7271	8.4583	9273.7285	0.0854	0.6086
	Im-Fpso–Otsu	96	1227.0154	8.3696	9465.0933	0.0830	0.6016
	FFA–Otsu	6	1217.3831	8.4715	9245.4802	0.0624	0.6040
	SFSMA–Otsu	7	1227.8124	8.4723	9243.7719	0.0791	0.6074

Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; FSIM, feature similarity index measure; Im-Fpso, fractional particle swarm optimization; MSE, mean squared error; PSNR, peak signal-to-noise ratio; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; SSIM, structural similarity index measure; WOA, Whale Optimization Algorithm.

On F9, both the MEAN and BEST values of SFSMA achieved the theoretical optimum, with a lower STD compared to other algorithms, demonstrating significantly better optimization stability than HFBOA and other counterparts. On F10, SFSMA performed similarly to PSO and SMA in key metrics, which could be attributed to the function's strong global optimum guidance, to which SMA's basic search mechanism was already well-suited. Nonetheless, SFSMA still maintained optimal MEAN and BEST values, while its STD was comparable to those of PSO and SMA. On F11, SFSMA, along with conventional algorithms and PSO, reached the theoretical optimum. However, SFSMA exhibited superior stability and a lower STD than both conventional algorithms and HFBOA. For F12, which exhibited multi-scale characteristics, SFSMA achieved the best MEAN and the smallest STD, leading in overall performance, whereas HFBOA was unable to match its optimization accuracy and stability on this function. Together, these results highlight the advantages of SFSMA in optimizing complex fixed multimodal functions and further validate the innovative value of the proposed algorithms.

5. Experimental results and analysis

To verify the overall performance of the SFSMA–Otsu, a multi-source heterogeneous image dataset was selected for this study to compare the threshold segmentation. The 2-D Otsu threshold segmentation method was combined with seven algorithms; the resulting approaches were as follows: SMA–Otsu; PSO–Otsu; BOA–Otsu; WOA–Otsu; Fractional PSO (Im-Fpso) proposed by Wei et al., [18] denoted as Im-Fpso–Otsu; FFA presented in Kong, [24] denoted as FFA–Otsu; and SFSMA–Otsu. The components of the experimental setup were: a central processing unit (Intel Core i9-13900HX CPU, Intel Corporation, USA), 16 GB of random-access memory, and a graphics processing unit (NVIDIA GeForce RTX 4060 GPU,

Figure 3. Human image segmentation results of seven algorithms. The original image was obtained from the Berkeley image database. [25] The differences in image segmentation performance among the seven algorithms are marked with red boxes

Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

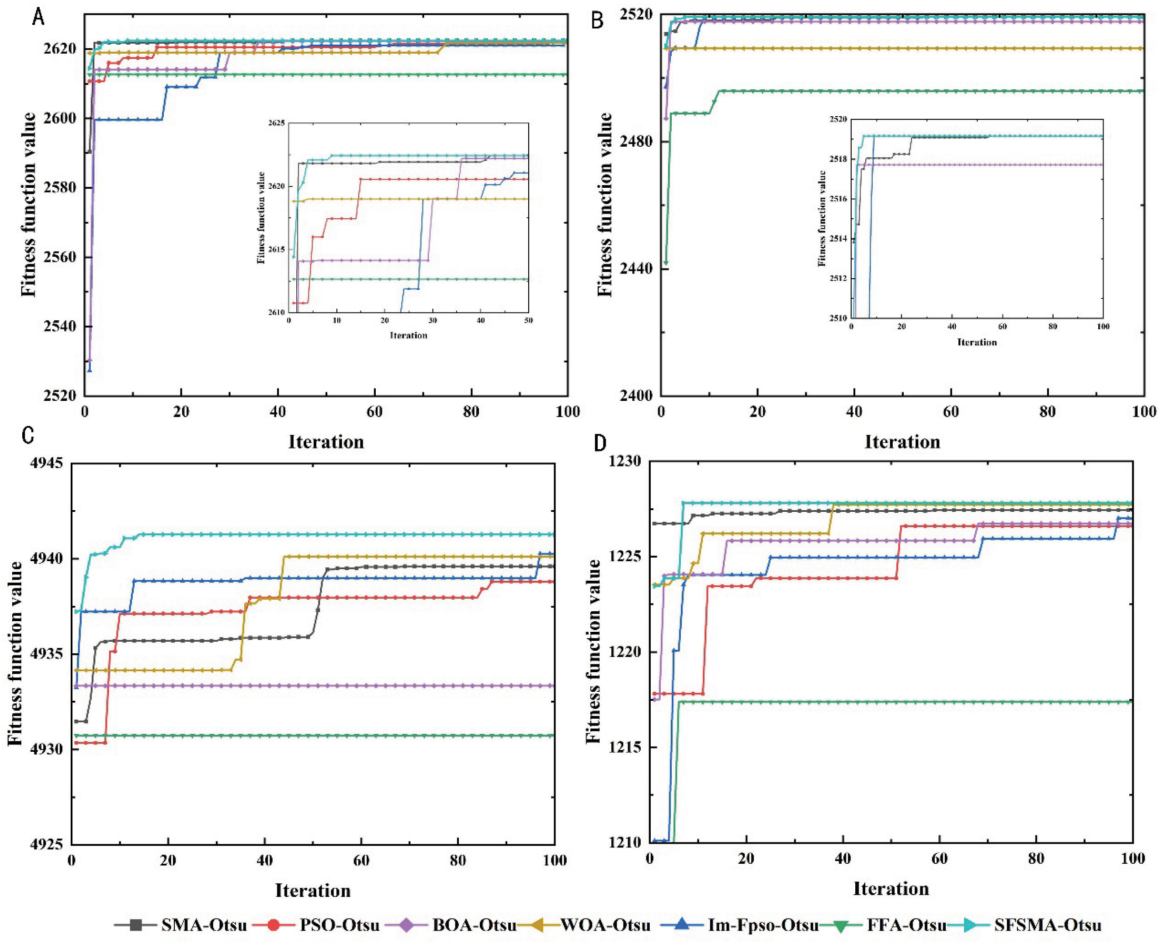

Figure 4. The fitness curves of human images. (A) Fitness curve of Lena's image. (B) Fitness curve of the blonde woman image. (C) Fitness curve of the pirate image. (D) Fitness curve of a couple image Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

NVIDIA Corporation, USA). The software environment was MATLAB (R2021a, MathWorks, Inc., USA). All algorithms employed unified parameter settings. The population size was set to 30 for all algorithms, with a maximum iteration count of 100.

This experiment adopted a multi-source heterogeneous image dataset containing human figures, landscapes, and medical images. Test images were selected from the Berkeley image database. Algorithms were objectively assessed based on convergence performance and segmentation precision, with the following evaluation metrics: fitness function, iteration count, PSNR, MSE, SSIM, and FSIM. A higher fitness value indicates that an algorithm can distinctly separate the background from the target, while fewer iterations required for convergence reflect faster computational speed. Higher PSNR values indicate stronger interference resistance; lower MSE values indicate greater similarity between segmentation results and the original image; and SSIM values closer to 1 indicate better preservation of details after segmentation. Image segmentation quality depends on the values of these indices.

The human image set included four samples: Lena, Woman-blonde, Pirate, and Couple. The original human images and their corresponding segmentation results are presented in Figure 3.

As shown in Figure 3, the SFSMA–Otsu yielded precise segmentation results for the brim texture and window details of the Lena test image. In the blonde-woman image, the SFSMA–Otsu was significantly superior to other algorithms in terms of segmentation completeness and clarity of hand contours, especially in maintaining better continuity in the knuckle regions. For the pirate image, more details

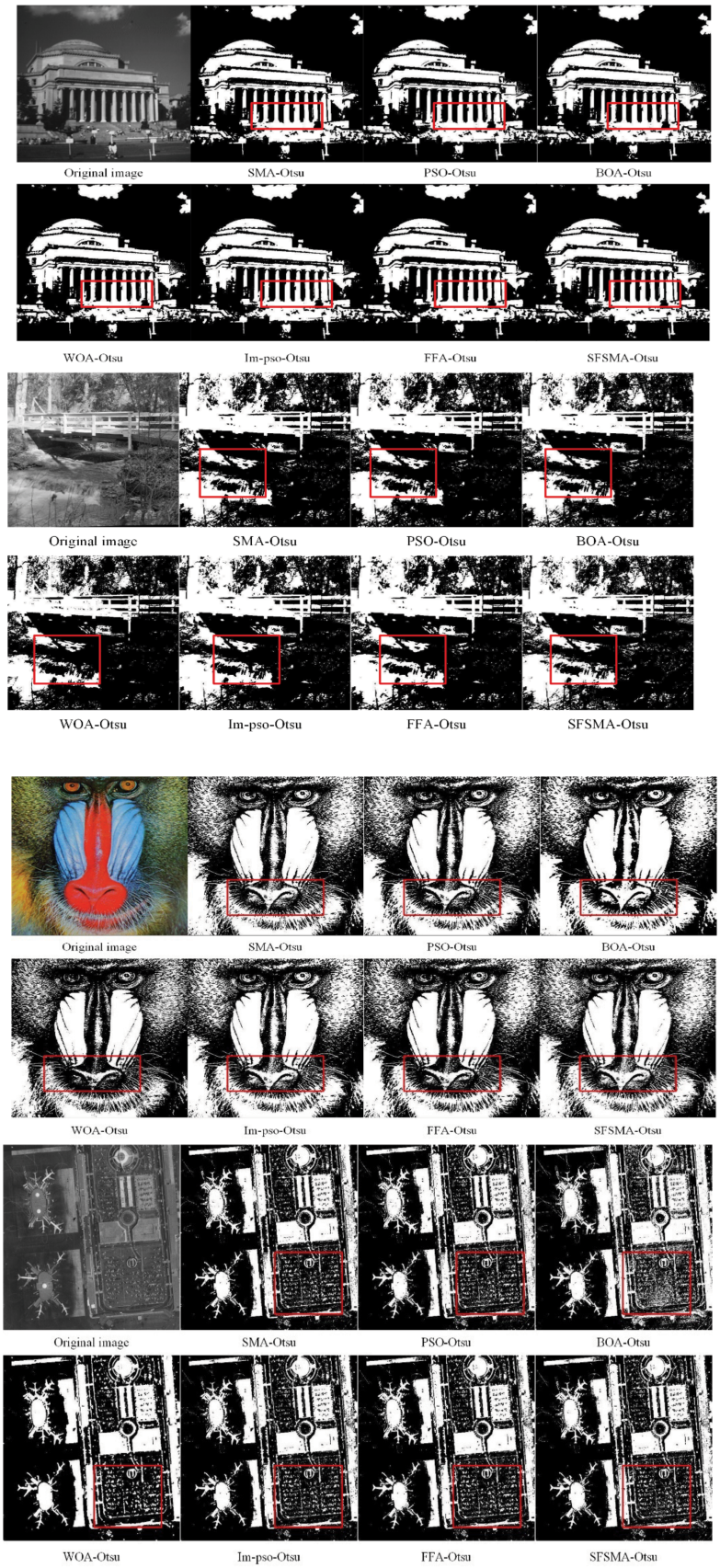

Figure 5. Landscape image segmentation results of seven algorithms. The original image was obtained from the Berkeley image database. [25] The differences in image segmentation performance among the seven algorithms are marked with red boxes
Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

Table 5. Objective evaluation indices corresponding to the landscape image segmentation results

Image	Algorithm	Iterations	Fitness	PSNR	MSE	SSIM	FSIM
	SMA–Otsu	46	2787.4668	9.9269	6612.8341	0.1091	0.4205
	PSO–Otsu	96	2786.9360	9.9652	6554.8060	0.1122	0.4203
	BOA–Otsu	70	2784.5761	9.9412	6591.1670	0.1144	0.4197
Building	WOA–Otsu	46	2787.2187	9.9412	6591.1670	0.1144	0.4197
	Im-Fpso–Otsu	5	2786.7056	9.9246	6616.3697	0.1074	0.4248
	FFA–Otsu	3	2783.2071	9.9449	6585.5042	0.1007	0.4217
	SFSMA–Otsu	9	2787.4668	9.9665	6552.8221	0.1093	0.4225
	SMA–Otsu	42	3840.3864	9.6391	7065.9323	0.1272	0.3688
	PSO–Otsu	69	3841.7894	9.5942	7139.3938	0.1187	0.3682
	BOA–Otsu	53	3844.8961	9.6395	7065.2848	0.1272	0.3684
Bridge	WOA–Otsu	2	3764.4562	9.5418	7226.0642	0.1304	0.3599
	Im-Fpso–Otsu	11	3843.6431	9.6201	7096.9539	0.1231	0.3689
	FFA–Otsu	1	3837.3139	9.5685	7181.7536	0.1116	0.3699
	SFSMA–Otsu	5	3840.3864	9.6608	7030.6583	0.1352	0.3693
	SMA–Otsu	58	1982.4250	8.7532	8664.8263	0.2236	0.5207
	PSO–Otsu	64	1981.9394	8.6938	8784.1592	0.2210	0.5123
	BOA–Otsu	47	1954.7816	8.5580	9063.0948	0.1769	0.5170
Mandril	WOA–Otsu	2	1941.4587	8.6426	8888.2869	0.2177	0.5470
	Im-Fpso–Otsu	84	1986.7971	8.7158	8681.3370	0.2131	0.5157
	FFA–Otsu	2	1979.1878	8.6825	8806.9526	0.2077	0.5198
	SFSMA–Otsu	7	1982.4250	8.7543	8662.5971	0.2238	0.5219
	SMA–Otsu	32	1489.7787	9.1821	7850.0568	0.1689	0.6118
	PSO–Otsu	42	1489.1336	9.1492	7909.7069	0.1734	0.6122
	BOA–Otsu	2	1437.9730	8.6947	8782.3564	0.1852	0.6178
Remote-sensing	WOA–Otsu	2	1455.6419	8.9068	8363.7790	0.1274	0.6080
	Im-Fpso–Otsu	51	1488.3788	9.1819	7850.3820	0.1609	0.6126
	FFA–Otsu	4	1489.1437	9.1596	7890.7581	0.1622	0.6131
	SFSMA–Otsu	11	1489.7787	9.1821	7850.0567	0.1689	0.6196

Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; FSIM, feature similarity index measure; Im-Fpso, fractional particle swarm optimization; MSE, mean squared error; PSNR, peak signal-to-noise ratio; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; SSIM, structural similarity index measure; WOA, Whale Optimization Algorithm.

were preserved in the segmentation results of the SFSMA–Otsu. For the couple image, SFSMA–Otsu yielded higher precision in segmenting curtain folds and carpet protrusions in the segmentation results of the SFSMA–Otsu compared to other algorithms. The differences in image segmentation performance among the seven algorithms are marked with red boxes.

Table 4 presents objective indices for human image segmentation outcomes, with fitness curves of these images displayed in Figure 4. Focusing on Lena image, Table 4 and Figure 4A show that the optimal segmentation threshold is achieved by the SFSMA–Otsu in five iterations, outperforming SMA–Otsu (42 iterations), PSO–Otsu (97 iterations), BOA–Otsu (36 iterations), WOA–Otsu (75 iterations), and Im-Fpso–Otsu (11 iterations), and confirming that its convergence is faster. Additionally, SFSMA–Otsu achieved superior fitness and PSNR values, as well as lower MSE compared to other methods. This result indicates that not only does the algorithm achieve quicker convergence, but it also achieves the highest segmentation accuracy, minimal distortion, and optimal image quality. It effectively resisted interference when locating global or near-global optimal thresholds and demonstrated clear advantages in both segmentation precision and efficiency. Table 4 and Figure 4B display the fitness curve and objective evaluation metrics for the blonde-woman image. For this image, the SFSMA–Otsu converged within five iterations. Its PSNR and MSE values were optimal, while its fitness and SSIM metrics surpassed those of competing algorithms. This shows that the SFSMA–Otsu not only fulfills its optimization goals effectively but also excels at assessing structural similarity between segmentation outputs and the original image. For the pirate image, compared to the SMA–Otsu, PSO–Otsu, BOA–Otsu, and Im-Fpso–Otsu, SFSMA–Otsu achieved

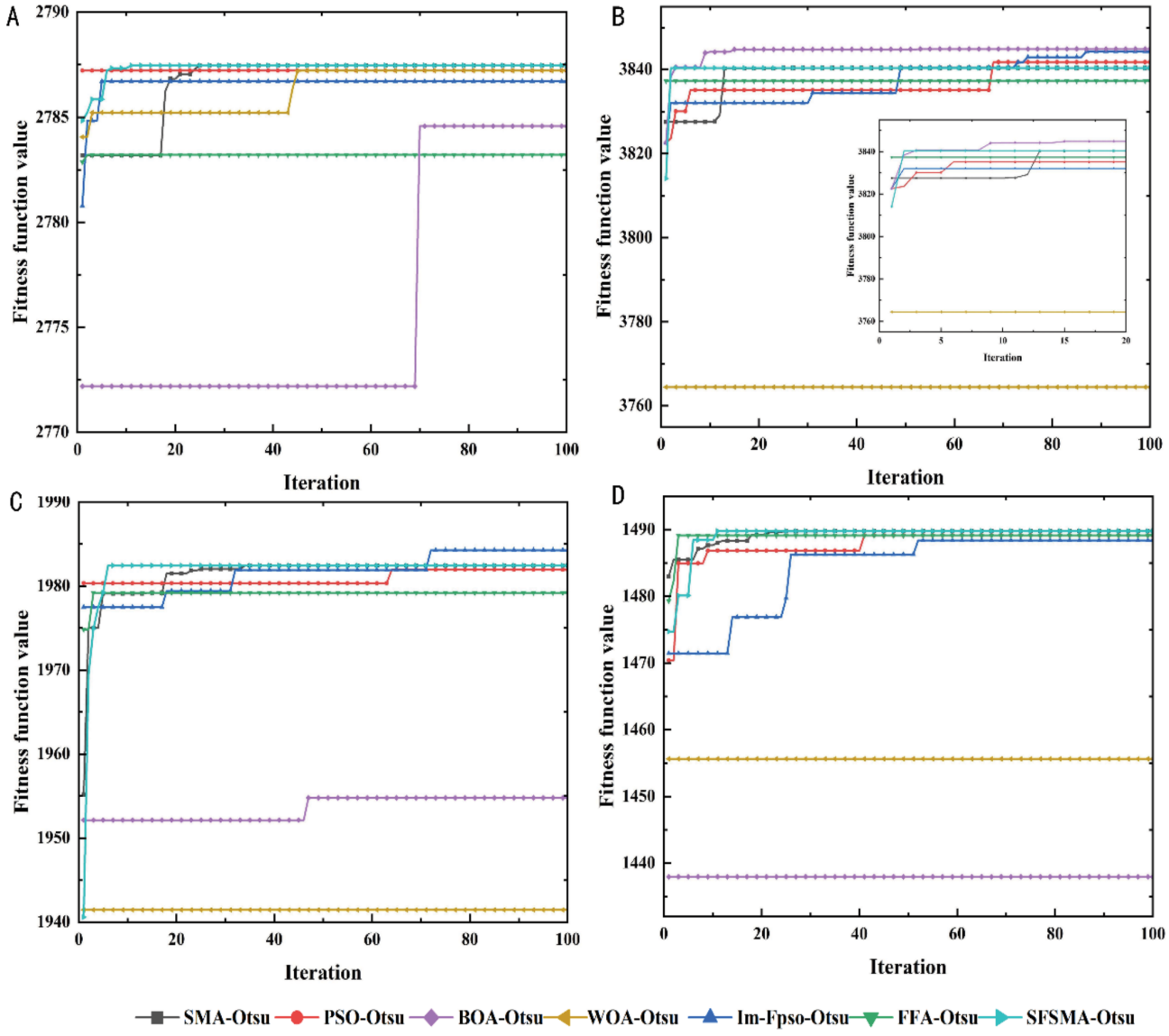

Figure 6. The fitness curves for the landscape image. (A) Fitness curve of building image. (B) Fitness curve of the bridge image. (C) Fitness curve of the mandril image. (D) Fitness curve of a remote-sensing image Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

speed improvements of 81.16%, 85.06%, 70.45%, and 63.89%, respectively. It also achieved the best results in terms of fitness, PSNR, and MSE values, with a significant SSIM score, outperforming most other algorithms. For the couple image, the SFSMA–Otsu converged in seven iterations, achieving the highest PSNR value, the lowest MSE value, and also performed well in terms of SSIM value. The FFA–Otsu consistently exhibited rapid convergence yet suboptimal metrics across segmentation tasks performed on the Lena, pirate, and couple images. As illustrated in the fitness curves in Figures 3 and 4, this behavior is attributed to the algorithm's tendency to become trapped in local optima during early iterations. Although rapid convergence was observed, the global optimum was not attained, thereby limiting overall segmentation performance. In summary, the SFSMA–Otsu yielded better segmentation results and faster segmentation efficiency.

The landscape image set included four samples: building, bridge, mandril, and remote-sensing. The original landscape images and their corresponding segmentation results are presented in Figure 5.

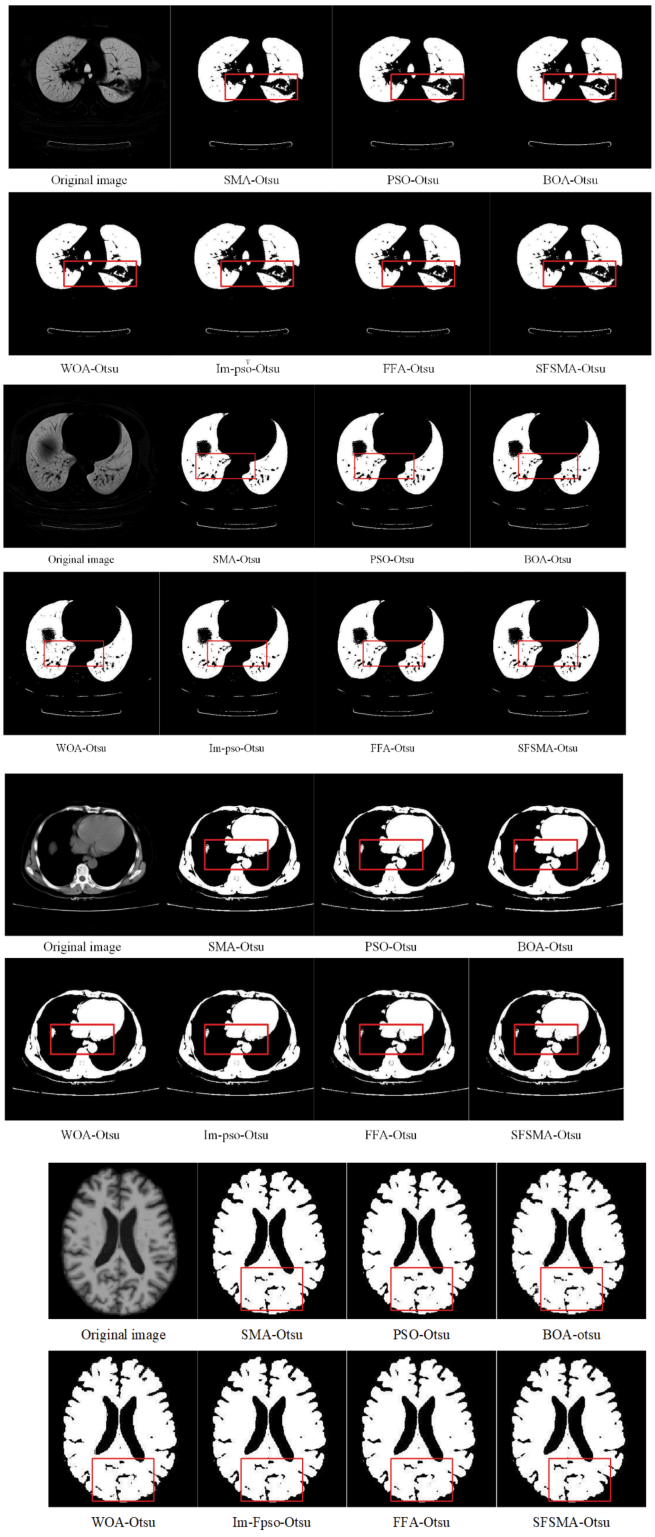

Figure 7. Medical image segmentation results of seven algorithms. The original image was obtained from the CT-GAN image database. [26] The differences in image segmentation performance among the seven algorithms are marked with red boxes

Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

Table 6. Objective evaluation indices corresponding to medical image segmentation results

Image	Algorithm	Iterations	Fitness	PSNR	MSE	SSIM	FSIM
Lung1	SMA–Otsu	41	4191.3482	14.8002	2153.0772	0.3107	0.7779
	PSO–Otsu	92	4191.3087	14.8221	2142.2466	0.3109	0.7792
	BOA–Otsu	93	4189.7136	14.8028	2151.8124	0.3078	0.7746
	WOA–Otsu	18	4190.9134	14.7804	2162.9268	0.3089	0.7773
	Im-Fpso–Otsu	5	4191.0334	14.8002	2135.6662	0.3114	0.7792
	FFA–Otsu	2	4045.8777	14.7879	2153.0880	0.3097	0.7702
	SFSMA–Otsu	12	4191.3482	14.8244	2141.1155	0.3110	0.7793
Lung2	SMA–Otsu	44	2948.8212	12.9858	3269.6471	0.3136	0.7466
	PSO–Otsu	59	2948.4147	12.9997	3259.1923	0.3130	0.7461
	BOA–Otsu	28	2948.3403	12.9773	3276.0840	0.3124	0.7457
	WOA–Otsu	4	2946.0397	12.9138	3324.3332	0.3174	0.7601
	Im-Fpso–Otsu	77	2948.4766	12.9674	3283.5039	0.3135	0.7478
	FFA–Otsu	29	2948.7340	13.0014	3257.9377	0.3135	0.7466
	SFSMA–Otsu	2	2948.8212	13.0185	3245.0956	0.3139	0.7468
CT	SMA–Otsu	67	4253.1521	13.1933	3117.1274	0.6360	0.8457
	PSO–Otsu	84	4252.7621	13.1962	3114.9857	0.6362	0.8464
	BOA–Otsu	2	4238.4053	13.0301	3236.4704	0.6215	0.8309
	WOA–Otsu	38	4245.3606	12.9874	3268.4690	0.6366	0.8464
	Im-Fpso–Otsu	50	4252.4497	13.1636	3138.5077	0.6364	0.8463
	FFA–Otsu	39	4252.5143	13.3096	3034.7591	0.6365	0.8463
	SFSMA–Otsu	12	4253.1522	13.2378	3085.3400	0.6366	0.8466
MRI	SMA–Otsu	43	7402.8584	9.6545	7040.9331	0.4132	0.6850
	PSO–Otsu	76	7402.2797	9.6400	7064.4444	0.4116	0.6830
	BOA–Otsu	2	7396.9836	9.5727	7174.8095	0.4147	0.6844
	WOA–Otsu	2	7399.0846	9.6535	7042.4852	0.4176	0.6873
	Im-Fpso–Otsu	5	7402.5924	9.6716	7013.3031	0.4129	0.6851
	FFA–Otsu	81	7401.9166	9.6222	7093.5460	0.4131	0.6853
	SFSMA–Otsu	9	7402.8584	9.6965	6973.1391	0.4140	0.6861

Abbreviations: BOA, Butterfly Optimization Algorithm; CT, computed tomography; FFA, Fractional Firefly Algorithm; FSIM, feature similarity index measure; Im-Fpso, fractional particle swarm optimization; MRI, magnetic resonance imaging; MSE, mean squared error; PSNR, peak signal-to-noise ratio; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; SSIM, Structural similarity index measure; WOA, Whale Optimization Algorithm.

As shown in Figure 5, SFSMA–Otsu achieved precise segmentation of domes and columns in building images. In the bridge image, significantly superior segmentation completeness and clarity of bridge contours (compared to other algorithms) were demonstrated by the SFSMA–Otsu, particularly at the junction where the bridge meets the river, where better visual effects were achieved. For mandril images, SFSMA–Otsu preserved more details of hair and skin textures in the segmentation compared to other algorithms. For remote-sensing images, it also retained additional details of ground-based facilities in its segmentation results compared to other algorithms. The differences in image segmentation performance among the seven algorithms are marked with red boxes.

In the building image segmentation, data from Table 5 and the fitness curve in Figure 6A demonstrate that SFSMA–Otsu converged within nine iterations, obtaining optimal fitness, PSNR, and MSE values. The FSIM value of 0.4225 surpassed most competing algorithms, including SMA (0.4205) and PSO (0.4203), though it was slightly lower than Im-Fpso's 0.4248. In comparison, algorithms such as SMA–Otsu and PSO–Otsu required over 46 iterations to converge. For the bridge image segmentation, SFSMA–Otsu converged in only five iterations, yielding the highest PSNR and lowest MSE, with optimal SSIM. The FSIM value of 0.3693 remained highly competitive, marginally lower than FFA's 0.3699. For mandril image segmentation, the proposed algorithm converged within seven iterations and outperformed all compared methods across objective metrics. Its FSIM value of 0.5219 exceeded those of SMA (0.5207) and PSO (0.5123), though it was slightly lower than WOA's 0.5470. In the remote-sensing image segmentation task, SFSMA–Otsu delivered outstanding performance with optimal fitness, PSNR, and MSE values. The FSIM value of 0.6196 ranked the highest among all algorithms. Moreover, the convergence speed was improved by 65.63%, 73.81%, and 78.43% compared to SMA–Otsu, PSO–Otsu, and Im-Fpso–Otsu, respectively. In summary,

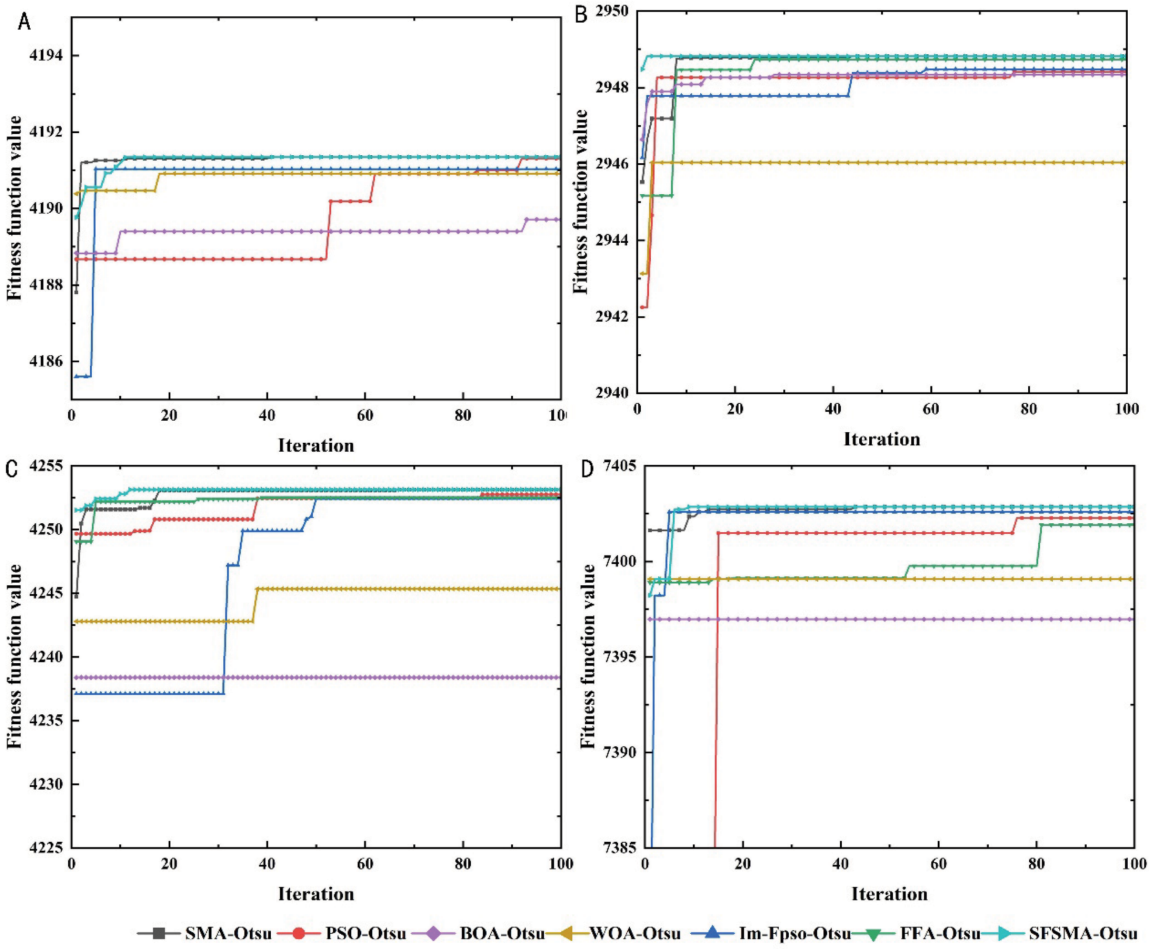

Figure 8. The fitness curves of medical images. (A) Fitness curve of Lung1 image. (B) Fitness curve of Lung2 image. (C) Fitness curve of the computed tomography image. (D) Fitness curve of a magnetic resonance image

Abbreviations: BOA, Butterfly Optimization Algorithm; FFA, Fractional Firefly Algorithm; Im-Fpso, fractional particle swarm optimization; PSO, particle swarm optimization; SFSMA, Strategy-based Fractional-order Slime Mould Algorithm; SMA, Slime Mould Algorithm; WOA, Whale Optimization Algorithm.

for landscape image segmentation, SFSMA–Otsu demonstrated significantly superior comprehensive performance, with faster convergence, better quantitative metrics, and clearer visual results compared to other algorithms.

The medical image set includes four samples: Lung1, Lung2, computed tomography (CT), and magnetic resonance imaging (MRI). The original medical images and their corresponding segmentation results are presented in Figure 7.

Figure 7 shows that while lung tissue can be fully segmented by the FFA–Otsu and Im-Fpso–Otsu, these two methods have limitations in detail preservation. In contrast, clearer segmentation results with more details were generated by the SFSMA–Otsu. For CT images, clearer thoracic structures—including the heart and lungs—were displayed in the segmentation outputs of the SFSMA–Otsu compared to other algorithms. In brain MRI images, brain structures were clearly outlined in this algorithm's segmentation results, enabling easy visualization of soft tissues. The differences in image segmentation performance among the seven algorithms are marked with red boxes.

Additional validation is provided by the performance metrics in Table 6 and the fitness curves in Figure 8. In the Lung1 image segmentation task, SFSMA–Otsu achieved optimal fitness, PSNR, and MSE values within 12 iterations, while maintaining strong SSIM performance. The FSIM value of 0.7793 was the highest overall, slightly exceeding the 0.7792 recorded by both PSO and Im-Fpso. For Lung2 image segmentation, the algorithm converged in only two iterations, achieving the best fitness, PSNR, and MSE results. The FSIM value of 0.7468 outperformed most compared algorithms, though it

was marginally lower than the 0.7601 achieved by WOA–Otsu. In CT image segmentation, SFSMA–Otsu attained optimal values across all evaluation metrics after 12 iterations, with an FSIM of 0.8466, the highest among all algorithms. The convergence efficiency was improved by over 68.42% compared to SMA–Otsu and other reference methods.

For MRI image segmentation, the proposed algorithm converged within nine iterations, whereas SMA–Otsu and other comparative algorithms required more than 43 iterations. SFSMA–Otsu achieved optimal fitness, PSNR, and MSE values, with an outstanding FSIM of 0.6861, only slightly lower than the 0.6873 obtained by WOA.

In conclusion, SFSMA–Otsu comprehensively outperformed all compared algorithms in medical image segmentation applications. The proposed modifications have substantially enhanced the performance of the algorithm.

6. Conclusion

We propose SFSMA to address the core limitations of the traditional SMA, including slow convergence, low optimization accuracy, and susceptibility to local optima. This enhanced algorithm integrated multiple strategies with fractional-order calculus to improve overall performance. To validate the effectiveness of SFSMA, a comprehensive experimental evaluation was conducted using 12 benchmark test functions spanning unimodal, multimodal, and high-dimensional complex types. Experimental results demonstrate that SFSMA achieved faster convergence speed, higher optimization accuracy, and enhanced robustness, confirming the effectiveness of the proposed improvements. To further extend the engineering applications and practical value of the algorithm, SFSMA was integrated with the 2-D Otsu threshold segmentation method, resulting in the SFSMA–Otsu segmentation algorithm. The proposed method was evaluated on a test set comprising four human, four landscape, and four medical images, with performance comparisons against six classical swarm intelligence-based segmentation algorithms. Quantitative experimental results show that SFSMA–Otsu achieved optimal comprehensive performance. Compared to the baseline SMA–Otsu, it demonstrated an average improvement of 0.2895% in PSNR, 0.6846% reduction in MSE, 0.2817% enhancement in SSIM, and 0.4064% increase in FSIM. These findings fully demonstrate the algorithm's enhanced robustness and practical utility, providing a reliable technical solution for complex optimization problems and image segmentation tasks.

Future studies should focus on enabling SFSMA to dynamically adjust its core parameters based on the characteristics of different optimization problems, thereby enabling it to adapt to a wider range of complex optimization scenarios. Additionally, SFSMA–Otsu should be extended to three-dimensional medical image segmentation, offering precise and efficient tools for medical image analysis.

Acknowledgments

None.

Funding

None.

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: Ruiyang Wu, Yu Ma
Formal analysis: Ruiyang Wu, Bowen Xiao
Investigation: Ruiyang Wu, Shoutong Huang
Methodology: Ruiyang Wu, Yu Ma
Writing–original draft: Ruiyang Wu
Writing–review & editing: Ruiyang Wu, Yu Ma

Availability of data

All data used in this research are publicly available without restrictions. The primary natural image dataset for segmentation algorithm evaluation is the Berkeley image database, retrievable via GitHub (https://github.com/BIDS/BSDS500.git, https://github.com/BIDS/BSDS300.git), Berkeley Institute for Data Science open links https://www2.eecs.berkeley.edu/Research /Projects/CS/vision/bsds/BSDS300-images.tgz, https://www2.eecs.berkeley.edu/Research/Projects /CS/vision/bsds/BSDS300-human.tgz), and IEEE Xplore (doi:10.1109/34.955137). The medical image dataset is the CT-GAN dataset, accessible through Kaggle (https://www.kaggle.com/ datasets/yisroelmirsky/ct-gan-medical-image-tampering) and the USENIX Security 19 conference repository.

AI tools statement

All authors confirmed that no AI tools were used in the preparation of this manuscript.

References

1. Liu XJ, Liu YL, Xu XX. Optimization of multi-threshold otsu image segmentation by glowworm swarm algorithm with cell membrane mechanism. *J Chin Comput Syst.* 2020;41(2):6. https://doi.org/CNKI:SUN:XXWX.0.2020-02-033
2. Bhandari AK, Kumar IV, Srinivas K. Cuttlefish algorithm-based multilevel 3-D Otsu function for color image segmentation. *IEEE Trans Instrum Meas.* 2020;69(5):2698-2708. https://doi.org/10.1109/TIM.2019.2922516
3. Dong Y, Li M, Zhou M. Multi-threshold image segmentation based on the improved dragonfly algorithm. *Mathematics.* 2024;12(6);854. https://doi.org/CNKI:SUN:JYRJ.0.2020-06-042
4. Ma Y, Ding Z, Zhang J, Ma Z. Otsu image segmentation algorithm based on hybrid fractional-order butterfly optimization. *Fractal Fract.* 2023;7(12):871. https://doi.org/10.3390/fractalfract7120871
5. Wang C, Tu C, Wei S, Yan L, Wei F. MSWOA: a mixed-strategy-based improved whale optimization algorithm for multilevel thresholding image segmentation. *Electronics.* 2023;12(12):2698. https://doi.org/10.3390/electronics12122698
6. Sowjanya K, Injeti SK. Investigation of butterfly optimization and gases brownian motion optimization algorithms for optimal multilevel image thresholding. *Expert Syst Appl.* 2021;182:115286. https://doi.org/10.1016/j.eswa.2021.115286
7. Sharifi T, Mirsalim M, Soleimanian Gharehchopogh F, Mirjalili S. Cultural history optimization algorithm: a new human-inspired metaheuristic algorithm for engineering optimization problems. *Neural Comput Appl.* 2025;37:21009-21068. https://doi.org/10.1007/s00521-025-11379-z
8. Abdel-Salam M, Houssein EH, Emam MM, Abdel Samee N, Soleimanian Gharehchopogh F, Bacanin N. EATHOA: elite-evolved hiking algorithm for global optimization and precise multi-thresholding image segmentation in intracerebral hemorrhage images. *Comput Biol Med.* 2025;196(Pt C):110835. https://doi.org/10.1016/j.compbiomed.2025.110835
9. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime Mould Algorithm: a new method for stochastic optimization. *Future Gener Comput Syst.* 2020; 111:300-323. https://doi.org/ 10.1016/j.future.2020.03.055
10. Chen X, Huang H, Heidari AA, et al. An efficient multilevel thresholding image segmentation method based on the Slime Mould Algorithm with bee foraging mechanism: a real case with lupus nephritis images. *Comput Biol Med.* 2022;142:105179. https://doi.org/10.1016/j.compbiomed.2021.105179
11. Houssein EH, Mahdy MA, Blondin MJ, Shebl D, Mohamed WM. Hybrid Slime Mould Algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst Appl.* 2021;174:114689. https://doi.org/10.1016/j.eswa.2021.114689
12. Guo YX, Liu S, Zhang L, Huang Q. Elite opposition-based learning quadratic interpolation Slime Mould Algorithm. *Appl Res Comput.* 2021;38(12):3651-3656. https://doi.org/10.19734/j.issn.1001-3695.2021.02.0175
13. Naik MK, Panda R, Abraham A. Normalized square difference based multilevel thresholding technique for multispectral images using leader Slime Mould Algorithm. *J King Saud Univ Comput Inf Sci.* 2022;34(7):4524-4536. https://doi.org/10.1016/j.jksuci.2020.10.030
14. Liu L, Zhao D, Yu F, et al. Performance optimization of differential evolution With Slime Mould Algorithm for multilevel breast cancer image segmentation. *Comput Biol Med.* 2021;138:104910. https://doi.org/10.1016/j.compbiomed.2021.104910
15. Hu J, Gui W, Heidari AA, et al. Dispersed foraging Slime Mould Algorithm: continuous and binary variants for global optimization and wrapper-based feature selection. *Knowl Based Syst.* 2022;237:107761. https://doi.org/10.1016/j.knosys.2021.107761
16. Pu YF. Application of fractional differential approach to digital image processing. *J Sichuan Univ (Eng Sci Ed).* 2007;(3):124-132. https://doi.org/10.1016/S1872-2075(07)60067-3
17. Li N, Hou X. Quantum image recognition for defect detection in logistics packaging. *Comput Eng Appl.* In press. 2025 [Epub ahead of print]. https://doi.org/CNKI:11.2127.tp.20250718.1341.012
18. Wei JR, Ma Y, Xia R, Jiang HB, Zhou TT. Otsu image segmentation algorithm based on fractional particle swarm. *Comput Eng Des.* 2017;38(12):3284-3290. https://doi.org/10.16208/j.issn1000-7024.2017.12.017
19. Fan Q, Ma Y, Wang P, Bai F. Otsu image segmentation based on a fractional order moth–flame optimization algorithm. *Fractal Fract.* 2024;8(2):24. https://doi.org/10.3390/fractalfract8020087
20. Wu YQ, Fan J, Wu SH. Fuzzy fractional order controller based on fractional calculus. *J Electron Meas Instrum.* 2011;25(3):218-225. https://doi.org/10.3724/SP.J.1187.2011.00218
21. Ren L, Heidari AA, Cai Z, et al. Gaussian Kernel probability-driven Slime Mould Algorithm with new movement mechanism for multi-level image segmentation. *Meas.* 2022;192:110884. https://doi.org/10.1016/j.measurement.2022.110884

22. Cao JY, Liang J, Cao BG. Research on fuzzy fractional-order controller based on fractional-order calculus. *J Xi'an Jiaotong Univ.* 2005;39(11):5. https://doi.org/10.3321/j.issn:0253-987X.2005.11.020
23. Chen QL, Huang G, Men T, Zhang XQ, Qin HY, Wang MR. Local fractional-order differential enhancement of digital images. *J Sichuan Univ (Eng Sci Ed).* 2016;48(4):115-122. https://doi.org/10.15961/j.jsuese.2016.04.016
24. Kong CY. *Research on Otsu Image Segmentation Algorithm Based on Firefly Algorithm Optimized by Fractional-order* [Doctoral dissertation]. Yinchuan: Ningxia University; 2019. https://doi.org/10.27257/d.cnki.gnxhc.2019.000046
25. Martin D, Fowlkes C, Tal D, Malik J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proc IEEE Int Conf Comput Vis. 2001;(ICCV 2001):416-423. doi:10.1109/ICCV.2001.937655
26. Mirsky Y, Mahler T, Shelef I, Elovici Y. CT-GAN. Malicious Tampering of 3D Medical Imagery using Deep Learning. Proc USENIX Secur Symp. 2019;(28th):arXiv:1901.03597v3 [cs.CR]. doi:10.48550/arXiv.1901.03597

Ruiyang Wu *Born in October 2000 in Yinchuan, Ningxia Province, P.R. China. She is currently a postgraduate student at the School of Electronic and Electrical Engineering, Ningxia University, majoring in Electronic Engineering.*
https://orcid.org/0009-0004-8942-7269

Yu Ma *Born in October 1974 in Yinchuan, Ningxia Province, P.R. China. He is a professor at Ningxia University and a senior member of the China Computer Society. His research interests include image processing, computer vision, and pattern recognition.*
http://orcid.org/0009-0005-8764-6035

Bowen Xiao *He is currently a postgraduate student at the School of Electronic and Electrical Engineering, Ningxia University, majoring in Electronic Science and Technology.*
https://orcid.org/0009-0007-0949-2389

Shoutong Huang *Born in August 2001 in Tianjin, P.R. China. She is currently a postgraduate student at the School of Electronic and Electrical Engineering, Ningxia University, majoring in Electronic Science and Technology.*
https://orcid.org/0009-0006-9560-3040

An International Journal of Optimization and Control: Theories & Applications
(https://accscience.com/journal/ijocta)