Renal Pathology Images Segmentation Based on Improved Cuckoo Search with Diffusion Mechanism and Adaptive Beta-Hill Climbing

Jiaochen Chen[1] · Zhennao Cai[1] · Huiling Chen[1] · Xiaowei Chen[2] · José Escorcia-Gutierrez[3] · Romany F. Mansour[4] · Mahmoud Ragab[5,6]

Abstract

Lupus Nephritis (LN) is a significant risk factor for morbidity and mortality in systemic lupus erythematosus, and nephro-pathology is still the gold standard for diagnosing LN. To assist pathologists in evaluating histopathological images of LN, a 2D Rényi entropy multi-threshold image segmentation method is proposed in this research to apply to LN images. This method is based on an improved Cuckoo Search (CS) algorithm that introduces a Diffusion Mechanism (DM) and an Adaptive β-Hill Climbing (AβHC) strategy called the DMCS algorithm. The DMCS algorithm is tested on 30 benchmark functions of the IEEE CEC2017 dataset. In addition, the DMCS-based multi-threshold image segmentation method is also used to segment renal pathological images. Experimental results show that adding these two strategies improves the DMCS algorithm's ability to find the optimal solution. According to the three image quality evaluation metrics: PSNR, FSIM, and SSIM, the proposed image segmentation method performs well in image segmentation experiments. Our research shows that the DMCS algorithm is a helpful image segmentation method for renal pathological images.

Keywords Multi-threshold image segmentation · 2D Rényi entropy · Renal pathology · Cuckoo search algorithm · Swarm intelligence algorithms · Bionic algorithm

1 Introduction

Renal pathology is a subspecialty of general pathology based on the diagnostic evaluation of light microscopy, immunofluorescence microscopy, and electron microscopy for medical and transplant kidney disease characteristics. Renal pathologists examine renal biopsies to provide a definitive clinicopathologic diagnosis, working closely with nephrologists, which is particularly significant for Lupus Nephritis (LN) [1]. LN is a significant cause of morbidity and mortality in systemic lupus erythematosus and has a broad histopathological spectrum [2, 3]. Renal biopsy is not necessary

✉ Zhennao Cai
cznao@wzu.edu.cn

✉ Huiling Chen
chenhuiling.jlu@gmail.com

Jiaochen Chen
21451943002@stu.wzu.edu.cn

Xiaowei Chen
179781679@qq.com

José Escorcia-Gutierrez
jescorci56@cuc.edu.co

Romany F. Mansour
romanyf@sci.nvu.edu.eg

Mahmoud Ragab
mragab@kau.edu.sa

[1] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

[2] Department of Rheumatology and Immunology, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, China

[3] Department of Computational Science and Electronics, Universidad de la Costa, CUC, 080002 Barranquilla, Colombia

[4] Department of Mathematics, Faculty of Science, New Valley University, 72511, El-Kharga, Egypt

[5] Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

[6] Department of Mathematics, Faculty of Science, Al-Azhar University, Naser City, Cairo 11884, Egypt

to confirm the diagnosis of LN but is essential to define the quality and severity of the renal injury and is a guide for treatment [4]. However, histopathological changes of LN are complex, varied, and sometimes unstructured; it needs expertise and effort to assess the biopsies. Nowadays, large amounts of digital image data from renal biopsies are being generated. Accordingly, there is a strong demand for developing computer-based image analysis systems for nephropathology, especially for the challenging images of LN.

Aiming development of intelligent medical systems [5], researchers apply more image processing techniques [6, 7]. The segmentation of kidney structure is an early step in computer-assisted nephropathological image analysis systems. In the early days, renal biopsies were manually visualized and interpreted by renal pathologists, but due to the need to reduce the consumption of time and resources and the support of science and technology, researchers began to seek ways to automate the process [8]. Albayrak et al. [9] proposed a two-stage image segmentation to obtain cellular structures in high-dimensional renal cell carcinoma histopathological images. In this method, a simple linear iterative clustering means is utilized to segment the image into superpixels, and then the superpixels are clustered by the latest segmentation method based on clustering. Experimental results show that a super-pixel segmentation algorithm can be used as a pre-segmentation method to improve the performance of cell segmentation.

Yoruk et al. [10] introduced a fully automated kidney segmentation technology for evaluating Glomerular Filtration Rate (GFR) in children, which improves iterative graph cut-based image segmentation methods by training a random forest classifier to segment kidney tissue into cortical, medullary and collecting systems. According to GFR, this automatic segmentation method has similar segmentation results to manual segmentation results to manual segmentation methods and saves much time, even reducing segmentation time from several hours to 45 s. Gadermayr et al. [11] proposed and studied a two-order pipeline consisting of weak supervision-based patch detection and accurate segmentation for glomerular segmentation. This pipeline does not require any previously acquired training data, thus reducing manual effort and manual annotation costs. Li et al. [12] put forward a novel graphic construction scheme that uses the optimal surface search method to solve, construct differences, and model the relationship between surfaces in the graph. The innovation of this graphic construction scheme is that it can simultaneously segment the outer and inner surface of the kidney, which solves the problem of renal cortex segmentation, which is rarely studied. Yang et al. [13] proposed a kidney partitioning method that achieves robust segmentation accuracy for dynamic contrast-enhanced MRI with little manual manipulation and parameter settings. The segmentation results match well with the manually generated

results, which is superior to the segmentation results of existing technical methods.

The work mentioned above mainly draws on machine learning and deep learning methods to achieve image segmentation; however, such methods have disadvantages, such as a long training cycle, large time and resource consumption, large data demand, and the inability of the trained model to be universal. The other image segmentation method, i.e., multi-threshold image segmentation technology, which is the most basic image segmentation technique, obtains appropriate thresholds by combining with swarm intelligence algorithms, which can solve the problems of large computation, long computation time, and low segmentation accuracy of traditional multi-threshold image segmentation methods [14]. In the last few years, this method has been recognized by many scholars and put into research due to its remarkable effect. Nguyen et al. [15] proposed a novel method for improving Moth Flame Optimization (MFO) [16] by mixing lévy flight and flame update formulations of logarithmic functions and proved that it has an excellent performance in the field of multi-threshold image segmentation. Zhou et al. [17] put forward a multi-threshold image segmentation technology based on the Moth Swarm Algorithm (MSA) and used Kapur's entropy method to optimize the threshold of the test image, and experiments verified the robustness and availability of the technique.

Zhao et al. [18] intensely studied the introduction of Horizontal Cross-Search (HCS) and Vertical Cross-Search (VCS) into a variant of Ant Colony Optimization (ACOR), improved the original ACOR mechanism, formed an improved algorithm called CCACO, and achieved a higher level of powerful results in image segmentation. Sharma et al. [19] proposed a new hybrid Butterfly Optimization Algorithm (BOA) called MPBOA by combining the BOA with the symbiotic and parasitic stages of the Symbiosis Organisms Search (SOS) algorithm. This method obtains the optimal threshold in the multi-threshold problem of single image segmentation, and the overall performance in search behavior and the convergence time is satisfactory. To increase the diversity of Whale Optimization Algorithm (WOA) solutions and avoid falling into local optimum, Chakraborty et al. [20] changed the random solution selection process in the search prey phase. They integrated the cooperative hunting strategy of whales into the exploitation phase of the algorithm. An Improved WOA (ImWOA) algorithm was proposed. The authors used the ImWOA method to effectively solve the multi-threshold image segmentation problem. Chakraborty et al. [21] also used the improved WOA algorithm to enhance the diagnostic efficiency of a computational tool that can quickly use COVID-19 chest X-ray photos to determine the severity of the disease. The improved WOA algorithm, called mWOAPR, integrates the random initialization population in the global search phase.

The COVID-19 chest X-ray images are segmented using the multi-threshold method and the Kapur's entropy fitness function calculated from the two-dimensional histogram of the gray image.

Bio-inspired computing is roughly defined as using computers to model the underlying mechanisms of biological phenomena and to study the use of computing from the perspective of intelligent populations [22]. Swarm intelligence-based computing is an essential branch of biological computing. To solve various problems, many computing algorithms based on swarm intelligence to simulate natural swarm models have been invented [23, 24]. The more well-known and standard swarm intelligence optimization algorithms involve Particle Swarm Optimization(PSO) [25], Differential Evolution(DE) [26], Gray Wolf Optimization(GWO) [27], Ant Colony Optimization(ACO) [28], Harris Hawks Optimization(HHO) [29], Whale Optimization Algorithm(WOA) [30], Genetic Algorithm(GA) [31], Cuckoo Search Algorithm(CS) [32] and Sine and Cosine Algorithm(SCA) [33] and so on. In addition, the swarm intelligence algorithm and its improved version have been effectively applied and developed in many aspects of life and production. Gharehchopogh et al. [34] introduced the lévy mutation operator, Cauchy mutation operator, and Gaussian mutation operator to improve the performance of the Tunicate Swarm Algorithm (TSA) to evaluate six large-scale engineering problems, named QLGCTSA algorithm. Tree Seed Algorithm (TSA) has been developed in hybridization, improvement, mutation, and optimization with its ability and advantages since it was proposed in 2015 [35, 36]. The Sparrow Search Algorithm (SSA) [37] is robust. Nowadays, its application in the structural improvement of artificial neural networks and the optimization of hyperparameter parameters of deep learning has also been recognized by scholars [38]. Since its introduction in 2014, the Symbiotic Organisms Search (SOS) has had improved versions, multi-objective versions, and hybrid discrete types, which are very effective in solving complex and NP-hard problems [39, 40]. WOA is a very popular optimization algorithm, and its improved versions are even more numerous. They play an important role in engineering, clustering, classification, robot path, image processing, network, task scheduling and other engineering applications [41]. The Moth Flame Optimization (MFO) algorithm is also a widely used optimization algorithm, and the butterfly optimization algorithm is a relatively new algorithm. Sahoo et al. combined the two and proposed a novel hybrid algorithm h-MFOBOA, which solves the two practical problems of optimal gas production capacity and three-bar truss design [42]. In besides, swarm intelligence algorithms have been applied to solve many other problems [21, 43–47].

Among the numerous swarm intelligence algorithms, the CS algorithm has only two control parameters, so it does not need to readjust every new problem. The algorithm is simpler and more general, has good scenarios in many fields, such as optimization and computational intelligence [48–52], and has excellent application scenarios. This paper uses the CS algorithm based on the original multi-threshold image segmentation technology to be improved. Following the no-free lunch theorem [53], no learning algorithm can consistently produce the most accurate learner in any field. Talking about which algorithm is better is meaningless when divorced from the actual situation. When dealing with optimization and multi-threshold image segmentation problems, the CS algorithm also has problems such as falling into local optimum and slowing convergence speed. Therefore, this paper introduces the Diffusion Mechanism (DM) and the Adaptive β-Hill Climbing (AβHC) mechanisms into the algorithm. DM strategy can generate more solutions by dispersing the population, increasing the cuckoo population diversity, and avoiding the algorithm from stagnation and local optimum. AβHC strategy makes the algorithm continuously explore more appropriately, contributing to the search for better solutions in the algorithm exploitation stage. In this article, the CS algorithm is chosen to be studied, and multi-threshold image segmentation experiments test the performance of the improved version. To verify the performance of the above strategies in the CS algorithm, the IEEE CEC2017 [54] dataset test experiment and renal pathology image segmentation experiment were carried out. The experimental results demonstrate that the proposed algorithm has improved its optimization ability to find the optimal solution for CEC2017 functions. The image segmentation effect is better according to the image quality evaluation metrics.

To summarize, the contributions of this article are as follows:

(1) An improved CS algorithm, DMCS, is proposed based on the DM and AβHC strategies.
(2) Combining the multi-threshold medical image segmentation method of renal pathology image with DMCS results in more accurate and higher-quality image segmentation results.
(3) Different image quality metrics are employed to assess the experimental results of image segmentation. The evaluation results show that the DMCS algorithm significantly improves its convergence and segmentation ability.

The rest of this paper is arranged as follows. Section 2 presents the proposed DMCS. Section 3 gives the experiment results and discussions on the CEC2017 dataset, and Sect. 4 offers the renal pathology image segmentation experiment. Finally, Sect. 5 summarizes the conclusions and future works.

2 Proposed DMCS

This part introduces the original CS algorithm and the proposed DMCS algorithm. Firstly, the rules and processes of the original CS algorithm are introduced. On this basis, the principles and mathematical formulas of the two mechanisms used in the DMCS algorithm and how to combine them with the original CS algorithm are introduced.

2.1 The Cuckoo Search (CS)

The CS algorithm is based on the parasitic brood behavior of cuckoo [32]. Furthermore, it is enhanced by Lévy flight instead of a simple random walk. Some experiments have proved that the CS algorithm, such as PSO, may be more efficient than others. Besides, the CS algorithm has been applied to many optimizations and computational intelligence fields, such as engineering design applications, data fusion, wireless sensor networks, thermodynamic calculation, image segmentation, etc. The CS algorithm summarizes the cuckoo's parasitic brood behavior into three rules [55]:

Each cuckoo will lay an egg at a time in a randomly placed nest.
The best nest with a good egg will be passed on to the next generation.
The number of host nests used is fixed, and the probability of a host finding the cuckoo egg is $0 \leq P_\alpha \leq 1$. If the host finds out that the egg in the nest is not its own, it can kill it or erect a new nest.

The search methods of the CS algorithm can be divided into two types: local search and global search. The switching parameter P_α controls the balance between the two search methods. Furthermore, local search is described by Eq. (1).

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(P_\alpha - \varepsilon) \otimes \left(x_j^t - x_k^t\right), \tag{1}$$

where x_i^t denotes the location of the i-th nest in the generation t, $\alpha > 0$ denotes the scaling factor of step size. s is the step size. $H(x)$ is the Heaviside function. x_j^t and x_k^t are two different solutions randomly selected in the t generation, ε is extracted from the uniform distribution, which is a random number. $\otimes$ means the dot product between two vectors.

The global search is performed by Lévy flight with Eq. (2).

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda), \tag{2}$$

where usually $\alpha = O\left(\frac{L}{10}\right)$, L is the feature scale of the problem in question; the value of α in Eqs. (1) and (2) may be different. To simplify the calculation, the value of α in two formulas is set to be the same. $L(s, \lambda)$ is a Lévy flight, which can be described by Eq. (3).

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi \lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}}, (s \geq s_0 > 0), \tag{3}$$

where $\Gamma(\lambda) = \int_0^\infty t^{z-1} e^{-t} dt$, and $0 \leq \lambda \leq 2$.

Given $r \in [0, 1]$, let r be compared with the switching probability P_α to get the global branch with Eq. (4).

$$x_i^{(t+1)} = \begin{cases} x_i^{(t)} & \text{if } r < P_\alpha \\ x_i^{(t)} + \alpha \otimes L(s, \lambda) & \text{if } r > P_\alpha \end{cases}. \tag{4}$$

Figure 1 illustrates the flowchart of the CS algorithm.

Fig. 1 The CS algorithm flowchart
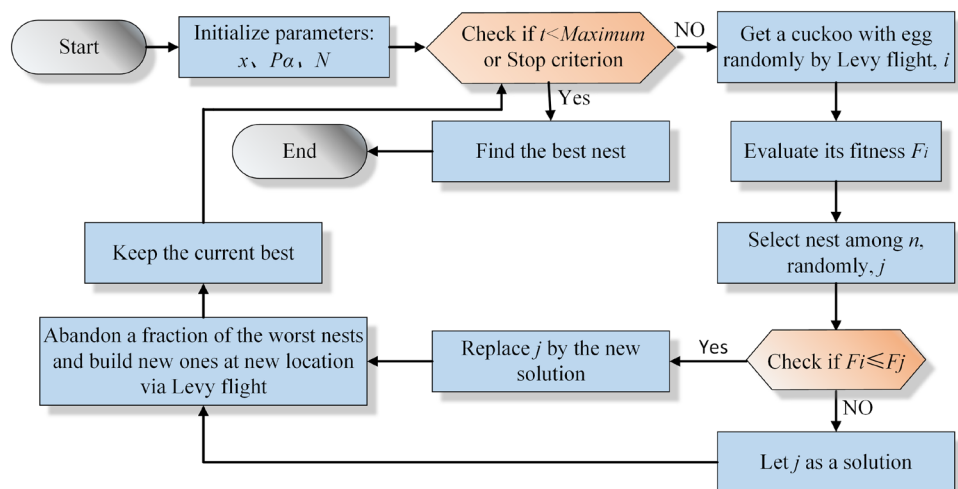
2.2 The Proposed DMCS

The original CS algorithm uses Lévy flight to update its agents' position. Although this strategy does an excellent job of searching global optima and jumping out from local optima, it still has some search space it cannot reach. Thus, some more suitable solutions may be missed. Here, the diffusion mechanism is used to address this shortcoming.

The exploration ability of the algorithm is enhanced by introducing the DM strategy into the CS algorithm. To further improve the algorithm exploitation capability, AβHC is introduced. AβHC is a simple and greedy local search algorithm. Although AβHC is not an intelligent optimization algorithm, it is a typical improvement strategy for other intelligent optimization algorithms. AβHC works after cuckoos seek new nests, using AβHC to find solutions with higher fitness values and thus develop better solutions.

2.2.1 The Diffusion Mechanism (DM)

The diffusion mechanism is the process by which the original particle randomly generates new particles around it using established strategies. The mathematical model of DM can be described by Eq. (5).

$$x_i = \text{Gaussian}\left(\text{Bestposition}, \varepsilon, m\right) + \sigma \times \left(\text{Bestposition} - x_i\right), \quad (5)$$

where x_i represents the position of a new particle generated by the diffusion strategy, and Gaussian(Bestposition, ε, m) denotes the generation of a random matrix obeying Gaussian distribution. *Bestposition* is the position of global optima, and σ is a random number obeying normal distribution, which is between 0 and 1. The m vector is a determinant of the form of the matrix generated.

2.2.2 Adaptive β-Hill Climbing (AβHC)

The adaptive β-hill-climbing algorithm [56] is an improved version of the β-hill-climbing algorithm [57], published in 2019 by Al-Betar et al. A detailed definition of this algorithm is presented below. Supposed that the mathematical description of the optimization problem is like this:

$$\min \left\{g(s) | s \in S\right\}, \quad (6)$$

where $s = \left\{s_1, s_2, \ldots, s_N\right\}$ denote a series of feasible solutions in the range of S. $S_i \in \left[LB_i, UB_i\right]$, LB_i and UB_i represent the lower and upper limits of the solutions severally. $g(s)$ is the objective function. N represents the population size.

Define the starting provisional solution to s_i. The solutions generated in each iteration are improved using the η − operator and β − operator. η − operator mainly play a role in the exploitation phase, using random walks for neighborhood search, and β − operator is similar to uniform mutation operator and mainly play a role in the exploration phase.

η − operator employs a concept called "random walk". The mathematical description using the η − operator can be described by Eq. (7):

$$s_i^{'} = s_i \pm U(0, 1) \times \eta, \quad (7)$$

where the current solution s_i can produce a new solution $s_i^{'}$ nearby by Eq. (7). $U(0, 1)$ denotes a random number with a normal distribution ranging from 0 to 1, and the distance between s_i and $s_i^{'}$ is determined by η. In the AβHC, η is an adaptive coefficient. The larger the value of η, the wider the search scope. Thus, to balance the exploitation and exploration phases, η needs a larger value to keep the algorithm searching more place in the early iterations and needs a small value to help the algorithm converge near the optimum. So it is reduced in each iteration from 1 to 0 according to Eq. (8):

$$\eta_t = 1 - \frac{t^{\frac{1}{p}}}{T^{\frac{1}{p}}}, \quad (8)$$

where t and T are the number of current iterations and the maximum number of iterations, respectively. p is a constant that brings the value of η closer to 0 as the number of iterations increases. This work set the value of p to 2. s', which is obtained by using η − operator to update the position, is used as the current solution, and the new position s'' gets further updated based on the current solution s' by using β − operator, as follows:

$$s_i^{''} = \begin{cases} s_k, r \leq \beta \\ s_i^{'}, \quad \text{otherwise} \end{cases}, \quad (9)$$

where r is a random number between 0 and 1, k is a random number selected from the range of i. Equation (9) states that if r is not bigger than β, then an individual from the current population will be randomly selected to replace the individual. Otherwise, it will remain unchanged. β is computed by a linearly increasing formula and depends on the $\beta_{\max}$ and $\beta_{\min}$. In the original paper, the author set $\beta_{\min}$ to 0.01 and $\beta_{\max}$ to 0.1. The formula is as follows:

$$\beta_t = \beta_{\min} + \frac{t}{T} \times \left[\beta_{\max} - \beta_{\min}\right]. \quad (10)$$

2.2.3 DMCS's Overall Framework

The proposed algorithm DMCS is dedicated to improving convergence speed and accuracy, and Algorithm 1 is the pseudo-code of DMCS. The DM mechanism is deployed innovatively to improve the algorithm convergence speed in the early iteration, while AβHC plays a role in improving the convergence accuracy of the algorithm in a later iteration. In addition, the flowchart of the DMCS is illustrated in Fig. 2.

The computational complexity of the DMCS algorithm is determined by the maximum generations $MaxFEs$, the number of individuals in the population N, and the dimensionality of individual dim. Thus, the computational complexity is $O(MaxFEs \times N \times dim)$.

algorithms on 30 benchmark functions of CEC 2017 to nine other advanced algorithms.

3.1 Experiment Setup

This subsection shows three experimental results on CEC 2017: the comparison experiments with the mechanisms, the traditional algorithm, and the advanced DMCS algorithm. These composition functions of the CEC 2017 are the combination of multiple shifted, rotation, and based multimodal functions. Therefore, they can broadly challenge the ability of the DMCS algorithm to solve real-world and complex optimization problems. We respected the universal rules of fair evaluations for all methods in this study [58–61]. These universal rules help researchers to ensure computing conditions are not biased, or some techniques do not perform superior only because of better conditions of tests [62,

---

Algorithm 1 The pseudo-code of DMCS

1: **begin**
2:    Objective function $f(x)$, $x = (x_1,...,x_d)^T$;
3:    Initialize the individual location x_i, $i = (1,2,...,n)$, the probability of finding p_α and
4:    the number of individuals N;
5:     **while** (t<Maximum Generation) or (stop criterion);
6:       Obtain a random cuckoo with egg (say i) by Lévy flight;
7:       Diffuse this nest by DM strategy and keep the best one (say i)
8:       Assess its quality/fitness; F_i
9:       Select a nest randomly among the n nests (say j);
10:       **if** ($F_i > F_j$)
11:         Use the new nest as j;
12:       **end if**
13:       Generate new nests by AβHC and replace the old nests;
14:       Abandon some of the poorer nests with a probability of P_a;
14:       and build new nests in a new location by Lévy flight;
15:       Retain nests with high-quality solutions;
16:       Rank the solutions to find the best current solution;
17:     **end while**
18:    Process results and graphing;
19: **end**

---

3 Experiments for Benchmark Functions Performance Testing

In this section, many experiments are done to assess the performance of the DMCS. To verify the effect of DM and AβHC on DMCS, the single strategy, and the combined strategy are tested on 30 benchmark functions from CEC 2017, respectively. Furthermore, to demonstrate that the DMCS algorithm has the ability of fast convergence, high convergence accuracy, and jumping out of local optimum, this paper compares DMCS with seven conventional

63]. To compare the experimental results more clearly, the results were treated with the Wilcoxon signed-rank test and the Freidman test [64]. The Friedman test is a nonparametric test of manifold judgment that can be utilized to evaluate the performance of different algorithms. Also, the lowest ordering is among the best-performing algorithms.

The population size is set as 30, and 30 independent runs are performed to test the algorithm's stability. Each test was carried out under the same conditions. The maximum function evaluation $MaxFEs$ is set at 30,000 for the mechanism comparison experiment and 300,000 for other experiments.
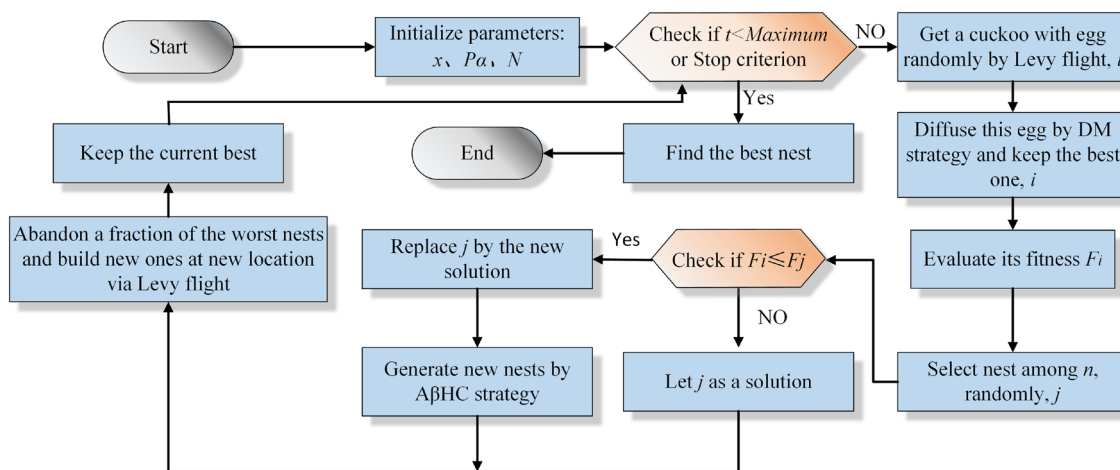
Fig. 2 Flowchart of the DMCS algorithm

All experiments were performed on the desk computer with Windows 10, Intel Core i5, 2.90 GHz, 8 GB RAM, and MATLAB R2021b.

3.2 Mechanism Comparison Experiment

To study the impact of the two mechanisms on the global optimization performance improvement of DMCS, the three improvement schemes combining the two mechanisms with CS were experimentally studied on the 30 benchmark test functions of CEC 2017 to find the best improvement scheme for CS. These three improved CS are denoted as MCS, DCS, and DMCS, respectively, as shown in Table 1. In the table, DM and AβHC represent the Diffusion Mechanism and the Adaptive β-hill-Climbing, respectively. "1" indicates that the algorithm uses this mechanism, and "0" indicates that it does not. For example, DCS uses the DM rather than the AβHC mechanism to improve.

Table 2 displays the Average (Avg), and Standard deviation (Std) of all algorithms generated by the experiments, where the best results are emboldened. The smaller the Avg, the better solution the algorithm finds on the benchmark functions. The smaller the Std, the more stable the algorithm is. Therefore, after adding the mechanism, the CS algorithm obtains better solutions and higher stability on more than 80% of the functions. Table 3 shows the results of the Wilcoxon signed-rank test analysis of the data in Table 2, where the best results are emboldened. The "mean" represents the average ranking results of each algorithm on all benchmark functions, and "rank" is the final ranking according to the "mean". Otherwise, " + /- / = " represents the number of corresponding performances of DMCS and other comparison algorithms on benchmark functions, where " + " denotes that DMCS performs better than the competitor, "-" denotes that DMCS performs worse than the competitor, " = " means that DMCS is similar to the competitor.

From Table 3, we can see that DMCS ranks first with an average value of 1.5, and DCS, and MCS rank second and third with average values of 1.9 and 2.43, respectively. DMCS, DCS, and MCS all rank ahead of CS; it can be seen that both strategies have a specific improvement on CS, and DM strategy plays a significant role. Figure 3 clearly shows the four algorithms' convergence speed and convergence accuracy. DM strategy makes a significant difference in accelerating the convergence speed of the DMCS algorithm. That is because DM can further strengthen the exploitation capability of the CS algorithm. Taking advantage of the characteristics of the diffusion process so that each cuckoo individual can spread around the current location to enhance the exploitation ability of the algorithm increases the chance of the algorithm finding the global optimum while also avoiding being trapped in the local optimum. The DM is used regularly during the function optimization process, enabling individuals in the algorithm to explore the search space more efficiently. Therefore, introducing the DM into the optimization algorithm can enhance the exploitation ability of CS and improve the algorithm's convergence rate and accuracy. AβHC strategy can improve the accuracy in the late stage.

3.3 Comparison with Conventional Algorithms

In this subsection, the proposed DMCS compared with seven conventional algorithms, namely Differential Evolution (DE)

Table 1 The combinatorial variants of CS

Algorithm	DM	AβHC
CS	0	0
MCS	0	1
DCS	1	0
DMCS	1	1

Table 2 The results achieved by combinatorial variants

Function	Criteria	DMCS	MCS	DCS	CS
F1	Avg	**1.46866E+06**	1.33351E+07	1.05665E+07	4.27375E+07
	Std	**7.94453E+05**	8.83944E+06	1.19943E+07	1.90345E+07
F2	Avg	**9.59788E+18**	8.06175E+24	1.40538E+20	7.39712E+25
	Std	**3.17714E+19**	3.34155E+25	3.98226E+20	2.02235E+26
F3	Avg	**4.46980E+04**	9.07989E+04	4.80161E+04	1.20048E+05
	Std	**9.19879E+03**	1.43283E+04	1.44958E+04	2.20881E+04
F4	Avg	5.34135E+02	**5.23068E+02**	5.23237E+02	5.58531E+02
	Std	4.84754E+01	**2.21722E+01**	3.03227E+01	2.35116E+01
F5	Avg	6.44815E+02	6.44101E+02	**6.36988E+02**	6.95354E+02
	Std	2.31547E+01	**2.26900E+01**	2.38971E+01	2.27934E+01
F6	Avg	**6.41297E+02**	6.42251E+02	6.43498E+02	6.54966E+02
	Std	**5.57602E+00**	9.23455E+00	5.83782E+00	6.76003E+00
F7	Avg	1.07585E+03	**9.48734E+02**	1.14480E+03	9.74957E+02
	Std	9.04760E+01	4.05282E+01	6.89698E+01	**4.01526E+01**
F8	Avg	9.87131E+02	**9.67270E+02**	9.71186E+02	9.81976E+02
	Std	4.63714E+01	2.79470E+01	3.70314E+01	**2.24837E+01**
F9	Avg	5.14595E+03	6.16837E+03	**5.09820E+03**	9.61942E+03
	Std	6.19988E+02	**1.89528E+03**	6.29417E+02	2.41748E+03
F10	Avg	**3.95687E+03**	4.90012E+03	4.38668E+03	5.44549E+03
	Std	5.18261E+02	3.17944E+02	3.40274E+02	**2.22643E+02**
F11	Avg	**1.32940E+03**	1.36732E+03	1.37897E+03	1.54328E+03
	Std	7.84130E+01	**5.92427E+01**	8.65915E+01	9.66773E+01
F12	Avg	3.11635E+06	4.65015E+06	2.34342E+06	**1.07955E+07**
	Std	4.27011E+06	3.51304E+06	**3.32474E+06**	5.48024E+06
F13	Avg	**2.79004E+03**	2.91868E+03	3.27981E+03	5.39405E+03
	Std	7.89108E+02	**4.73869E+02**	9.53197E+02	8.81380E+02
F14	Avg	**1.73346E+03**	1.78355E+03	1.88973E+03	2.09799E+03
	Std	**9.68741E+01**	1.15988E+02	1.87186E+02	2.15389E+02
F15	Avg	**2.84993E+03**	3.31661E+03	3.82931E+03	7.63819E+03
	Std	1.04317E+03	**6.67849E+02**	2.27051E+03	1.69161E+03
F16	Avg	**2.35900E+03**	2.62867E+03	2.48258E+03	2.84995E+03
	Std	1.83118E+02	2.49716E+02	2.53173E+02	**1.43592E+02**
F17	Avg	**2.17637E+03**	2.22562E+03	2.22942E+03	2.31461E+03
	Std	2.02810E+02	**1.06635E+02**	2.40861E+02	9.97462E+01
F18	Avg	7.88313E+04	1.30013E+05	**7.66129E+04**	2.62459E+05
	Std	**5.85879E+04**	7.60127E+04	6.83323E+04	1.34541E+05
F19	Avg	**2.08393E+03**	2.13320E+03	2.15183E+03	2.27264E+03
	Std	1.00497E+02	**6.91815E+01**	1.07351E+02	1.48096E+02
F20	Avg	**2.42971E+03**	2.58030E+03	2.46215E+03	2.66734E+03
	Std	1.20432E+02	1.20337E+02	**7.25603E+01**	1.01660E+02
F21	Avg	2.23166E+03	**2.22667E+03**	2.23965E+03	2.25479E+03
	Std	4.03909E+01	2.36630E+01	3.47190E+01	**1.34038E+01**
F22	Avg	2.39540E+03	2.37290E+03	**2.36111E+03**	2.39661E+03
	Std	4.51130E+01	**2.60207E+01**	3.30610E+01	3.17008E+01
F23	Avg	**2.50000E+03**	2.97894E+03	**2.50000E+03**	3.02370E+03
	Std	**0.00000E+00**	3.87429E+01	**0.00000E+00**	3.41191E+01
F24	Avg	**2.60000E+03**	3.22072E+03	**2.60000E+03**	3.24121E+03
	Std	**0.00000E+00**	3.81015E+02	**0.00000E+00**	1.94256E+02
F25	Avg	**2.70000E+03**	2.98822E+03	**2.70000E+03**	3.01994E+03
	Std	**0.00000E+00**	4.08643E+01	**0.00000E+00**	2.94468E+01

Table 2 (continued)

Function	Criteria	DMCS	MCS	DCS	CS
F26	Avg	**2.80000E+03**	4.86575E+03	**2.80000E+03**	5.70124E+03
	Std	**0.00000E+00**	1.19909E+03	**0.00000E+00**	6.20071E+02
F27	Avg	**2.90000E+03**	3.55747E+03	**2.90000E+03**	3.62485E+03
	Std	**0.00000E+00**	7.23919E+01	**0.00000E+00**	7.28939E+01
F28	Avg	**3.00000E+03**	3.43720E+03	**3.00000E+03**	3.53430E+03
	Std	**0.00000E+00**	4.12102E+02	**0.00000E+00**	1.22465E+02
F29	Avg	**3.10000E+03**	3.94114E+03	**3.10000E+03**	4.07349E+03
	Std	**0.00000E+00**	2.09364E+02	**0.00000E+00**	1.39809E+02
F30	Avg	**3.20000E+03**	1.30684E+05	**3.20000E+03**	2.49052E+05
	Std	**0.00000E+00**	5.27251E+04	**0.00000E+00**	1.16942E+05

Table 3 The results of Avg and Std analysis of the mechanism comparison experiment

F	DMCS	MCS	DCS	CS
F1	1	3	2	4
F2	1	3	2	4
F3	1	3	2	4
F4	3	1	2	4
F5	3	2	1	4
F6	1	2	3	4
F7	3	1	4	2
F8	4	1	2	3
F9	2	3	1	4
F10	1	3	2	4
F11	1	2	3	4
F12	2	3	1	4
F13	1	2	3	4
F14	1	2	3	4
F15	1	2	3	4
F16	1	3	2	4
F17	1	2	3	4
F18	2	3	1	4
F19	1	2	3	4
F20	1	3	2	4
F21	2	1	3	4
F22	3	2	1	4
F23	1	3	1	4
F24	1	3	1	4
F25	1	3	1	4
F26	1	3	1	4
F27	1	3	1	4
F28	1	3	1	4
F29	1	3	1	4
F30	1	3	1	4
+ / - / =	~	16/1/13	6/1/23	27/1/2
Mean	1.50	2.43	1.90	3.90
Rank	1	3	2	4

[26], Gray Wolf Optimizer (GWO) [27], Harris Hawks Optimization (HHO) [29], Slime Mould Algorithm (SMA) [65], Cuckoo Search (CS) algorithm [32], Multi-Verse Optimizer (MVO) [66] and Farmland Fertility Algorithm (FFA) [67].

Table 4 displays the Avg, and Std of all algorithms, where the best results are emboldened. DMCS obtained the smallest Avg and Std on F1, F2, F4, F12, F18, and F19, the smallest Std on F10 and F21, and tied with HHO for the best results on F23-F30. These indicate that DMCS has a significant advantage in both optimization ability and stability compared with the conventional algorithms in the experiment. Table 5 displays the results that analyze the data in Table 4 through the Wilcoxon signed-rank test, where the best results are emboldened. As shown in Table 5, DMCS gets the first in the "rank", in which DMCS ranks first in sixteen and second in five functions. It is clear that DMCS ranks best on more than 60% of the benchmark functions and significantly outperforms the competitors on at most 28 and at least 13 functions.

Moreover, Fig. 4 shows the Freidman test result of DMCS and its competitors. This figure shows that DMCS ranks NO.1 with a Freidman test value of 2.852, a clear advantage over the SMA, which ranks second with a mean value of 3.701, and even far below the other competitors. Figure 5 plots and shows the convergence curves of the methods for some functions when solving the benchmark functions of IEEE CEC 2017. It can also be seen that DMCS has higher convergence accuracy and faster convergence on most benchmark functions, which also shows that DMCS is easier to jump out of local optimization solutions and obtain a higher-quality global optimization solution.

3.4 Comparison with Advanced Algorithms

In this subsection, the performance of DMCS was further verified by comparing it with nine advanced algorithms on 30 benchmark functions from CEC 2017. They are Boosted Grey Wolf Optimizers (OBLGWO) [68], Modified Sine Cosine Algorithm (MSCA) [69], Hybridizing Gray
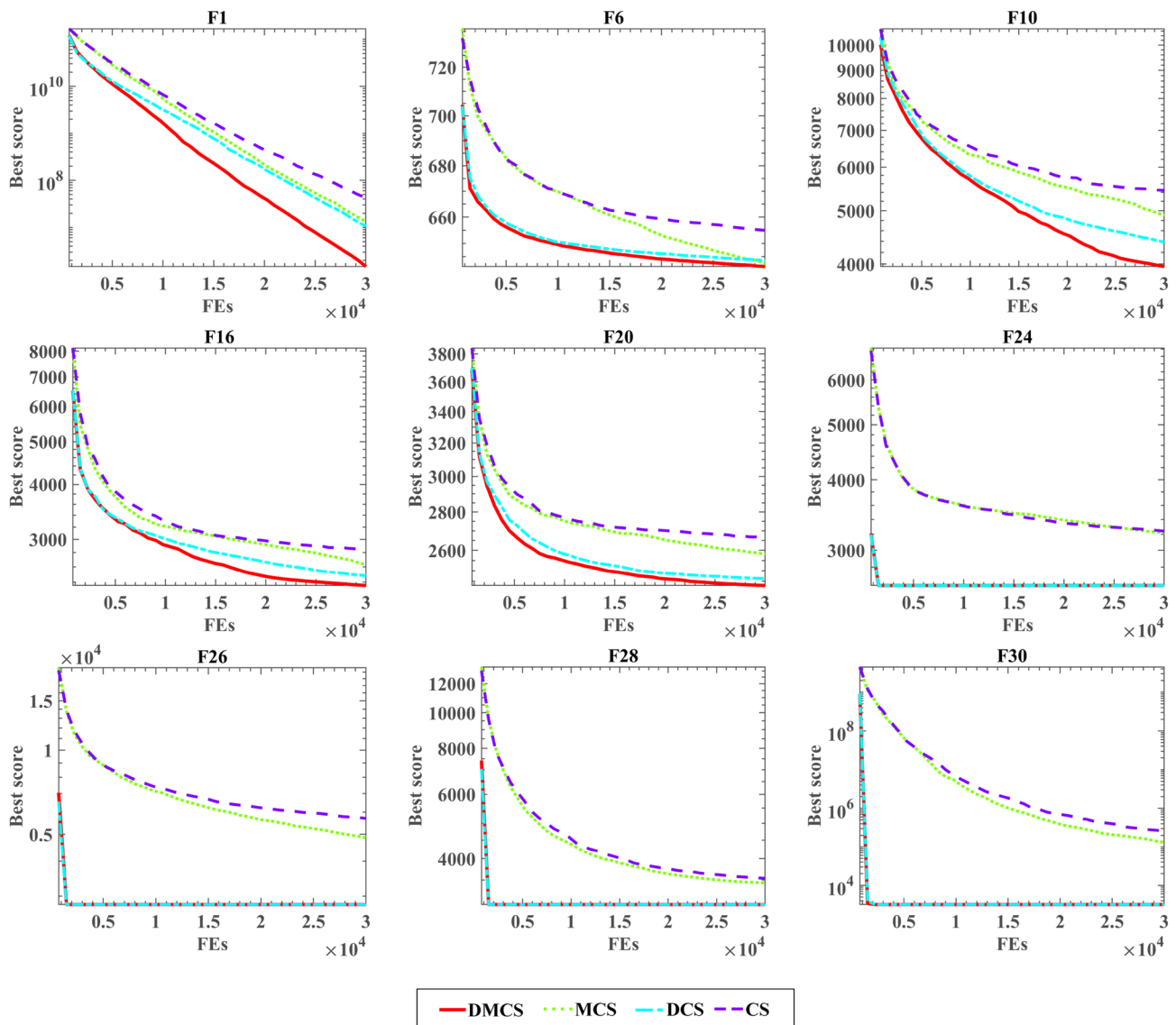
Fig. 3 The convergence curves of DMCS, MCS, DCS

Wolf Optimization with Differential Evolution (HGWO) [70], Orthogonal Learning-driven Multi-swarm Sine Cosine Algorithm (OMGSCA) [71], Comprehensive Learning Particle Swarm Optimizer (CLPSO) [72], Particle Swarm Optimization with an Aging Leader and Challengers (ALCPSO) [25], Lévy flight trajectory-based Whale Optimization Algorithm (LWOA) [73], Double Adaptive Random Spare Reinforced Whale Optimization Algorithm (RDWOA) [74], Gaussian Barebone Harris Hawks Optimizer (GBHHO) [75].

Table 6 displays the Avg and Std of all algorithms, where the best results are emboldened. The DMCS achieved the smallest Avg or Std on F1–F4, F13–F15, F18, F19, F21, and F23–F30, which indicates that DMCS is still intensely competitive with advanced algorithms.

Table 7 displays the results that further analyze the data in Table 6 through the Wilcoxon signed-rank test. Analyzing the Avg and Std obtained in the experiment, it can be concluded that the quality of the optimal solution obtained by the DMCS algorithm is of better quality and more stable for most functions. DMCS generally ranks first with an average of 2.7 in Table 7, first in 17 functions, and second in two cases, significantly better than the competition for at least 14 functions and significantly better than the LWOA algorithm for all 30 functions. Especially for several multimodal functions from F23 to F30, DMCS has breakneck convergence speed and high solution accuracy. Besides, Fig. 6 shows the bar graphs of the Friedman test result of DMCS and other competitors. From the graph of the result of the Friedman ranking, one conclusion can be

Table 4 The results achieved by DMCS against the seven conventional algorithms

Function	Criteria	DMCS	DE	HHO	WOA
F1	Avg	**1.00000E+02**	3.26384E+02	1.29537E+07	3.92240E+06
	Std	**2.61731E-11**	8.63988E+02	3.53825E+06	3.26431E+06
F2	Avg	**9.43453E+03**	2.62227E+25	1.27545E+13	9.44692E+26
	Std	**3.21925E+04**	4.84485E+25	6.78037E+13	3.83602E+27
F3	Avg	3.09710E+02	2.10805E+04	2.68273E+03	1.50007E+04
	Std	1.88720E+01	4.82665E+03	9.54007E+02	4.99572E+03
F4	Avg	**4.06782E+02**	4.94389E+02	5.79277E+02	5.86559E+02
	Std	**2.02458E+01**	3.35179E+01	6.28986E+01	5.38589E+01
F5	Avg	6.43704E+02	6.11826E+02	6.68148E+02	7.07512E+02
	Std	2.68512E+01	7.91988E+00	2.12728E+01	5.66895E+01
F6	Avg	6.29978E+02	**6.00000E+02**	6.52997E+02	6.63374E+02
	Std	7.95065E+00	**0.00000E+00**	3.53646E+00	9.30389E+00
F7	Avg	1.00973E+03	8.42237E+02	1.30846E+03	1.30904E+03
	Std	8.21536E+01	**9.34008E+00**	8.93223E+01	1.09090E+02
F8	Avg	9.51878E+02	**9.10893E+02**	1.07508E+03	1.09072E+03
	Std	3.48951E+01	**7.70761E+00**	4.46819E+01	6.92287E+01
F9	Avg	4.15386E+03	**9.00000E+02**	7.75765E+03	9.39778E+03
	Std	6.77502E+02	**2.11111E-14**	7.20713E+02	3.62527E+03
F10	Avg	**3.84894E+03**	5.64063E+03	4.70195E+03	5.83914E+03
	Std	4.20039E+02	**2.86375E+02**	6.08293E+02	6.40620E+02
F11	Avg	1.22552E+03	1.15291E+03	1.40674E+03	1.49670E+03
	Std	6.02212E+01	**8.82423E+00**	7.26056E+01	1.07634E+02
F12	Avg	**4.32651E+03**	9.50324E+03	6.33464E+07	1.31387E+08
	Std	**1.17263E+03**	5.89547E+03	5.58911E+07	6.02270E+07
F13	Avg	1.41153E+03	**1.34181E+03**	2.27955E+05	1.22719E+05
	Std	7.31360E+01	**8.78826E+00**	9.78269E+04	9.38442E+04
F14	Avg	1.47773E+03	**1.46370E+03**	4.55392E+04	2.92626E+05
	Std	3.98115E+01	**8.20058E+00**	4.32098E+04	1.56253E+05
F15	Avg	1.58878E+03	**1.54935E+03**	3.43142E+04	3.80375E+04
	Std	2.73438E+01	**1.33731E+01**	2.04783E+04	3.30425E+04
F16	Avg	2.25038E+03	**2.07445E+03**	3.05121E+03	3.20858E+03
	Std	2.24036E+02	**1.47870E+02**	3.19813E+02	3.36885E+02
F17	Avg	2.15688E+03	**1.94933E+03**	2.43364E+03	2.66424E+03
	Std	2.14400E+02	**3.22497E+01**	2.55771E+02	2.61422E+02
F18	Avg	**1.23233E+04**	2.84521E+05	1.28502E+06	5.59478E+06
	Std	**6.08886E+03**	1.40864E+05	1.61761E+06	6.52036E+06
F19	Avg	**1.93245E+03**	5.29756E+03	1.39012E+05	6.25499E+05
	Std	**1.03834E+01**	2.09573E+03	1.19443E+05	6.17882E+05
F20	Avg	2.39975E+03	**2.20879E+03**	2.74585E+03	2.69990E+03
	Std	1.24796E+02	**3.70127E+01**	2.36364E+02	2.02995E+02
F21	Avg	**2.12475E+03**	2.19540E+03	2.26046E+03	2.27338E+03
	Std	3.90522E+01	2.05806E+01	3.93877E+01	3.67739E+01
F22	Avg	2.36782E+03	2.31299E+03	2.42327E+03	2.42782E+03
	Std	3.46431E+01	**9.99721E+00**	2.51995E+01	3.66425E+01
F23	Avg	**2.50000E+03**	2.87024E+03	**2.50000E+03**	3.15768E+03
	Std	**0.00000E+00**	1.15338E+01	**0.00000E+00**	1.05748E+02
F24	Avg	**2.60000E+03**	3.39616E+03	**2.60000E+03**	2.63559E+03
	Std	**0.00000E+00**	6.82123E+00	**0.00000E+00**	1.94959E+02
F25	Avg	**2.70000E+03**	2.91143E+03	**2.70000E+03**	2.70000E+03
	Std	**0.00000E+00**	6.08754E+00	**0.00000E+00**	3.15962E-13

Table 4 (continued)

Function	Criteria	DMCS	DE	HHO	WOA
F26	Avg	**2.80000E+03**	5.40804E+03	**2.80000E+03**	4.05132E+03
	Std	**0.00000E+00**	8.73906E+01	**0.00000E+00**	2.34201E+03
F27	Avg	**2.90000E+03**	3.43276E+03	**2.90000E+03**	3.93945E+03
	Std	**0.00000E+00**	1.24627E+01	**0.00000E+00**	1.83197E+02
F28	Avg	**3.00000E+03**	3.98602E+03	**3.00000E+03**	3.09197E+03
	Std	**0.00000E+00**	7.70597E+02	**0.00000E+00**	1.56663E+02
F29	Avg	**3.10000E+03**	3.46953E+03	**3.10000E+03**	4.41127E+03
	Std	**0.00000E+00**	9.00783E+01	**0.00000E+00**	3.74846E+02
F30	Avg	**3.20000E+03**	6.76076E+04	**3.20000E+03**	2.48164E+06
	Std	**0.00000E+00**	2.24925E+04	**0.00000E+00**	1.84860E+06
Function	Criteria	GWO	SMA	MVO	FFA
F1	Avg	3.18940E+09	4.82590E+06	1.32305E+04	1.56500E+02
	Std	3.11286E+09	4.68053E+06	2.24939E+04	1.12613E+02
F2	Avg	5.87555E+28	2.43668E+18	2.73194E+04	8.46338E+16
	Std	1.02202E+29	1.26841E+19	6.47684E+04	4.07871E+17
F3	Avg	3.16563E+04	3.93229E+03	**3.00341E+02**	9.22898E+03
	Std	9.23329E+03	1.95358E+03	**1.51509E-01**	2.49999E+03
F4	Avg	6.29035E+02	5.13006E+02	4.82601E+02	4.58904E+02
	Std	8.98745E+01	2.94747E+01	3.63401E+01	4.36412E+01
F5	Avg	5.80510E+02	6.05920E+02	5.91990E+02	**5.74232E+02**
	Std	**1.57831E+01**	1.86882E+01	2.09102E+01	4.27661E+01
F6	Avg	6.05815E+02	6.18986E+02	6.07878E+02	**6.00000E+02**
	Std	2.35832E+00	7.64346E+00	7.11632E+00	2.19183E-05
F7	Avg	8.93813E+02	9.29221E+02	8.33897E+02	**8.30572E+02**
	Std	5.32182E+01	4.39901E+01	3.09393E+01	5.43303E+01
F8	Avg	8.92198E+02	9.18735E+02	8.86798E+02	8.90739E+02
	Std	2.43725E+01	3.03645E+01	2.35300E+01	5.03139E+01
F9	Avg	2.36638E+03	4.05672E+03	2.43746E+03	9.06638E+02
	Std	8.72449E+02	8.27642E+02	2.15259E+03	1.16075E+01
F10	Avg	3.85245E+03	3.97140E+03	3.92417E+03	7.00943E+03
	Std	4.47613E+02	4.10119E+02	7.37215E+02	9.27193E+02
F11	Avg	3.03537E+03	1.24522E+03	1.27224E+03	**1.14402E+03**
	Std	2.06130E+03	3.46774E+01	5.53795E+01	2.77462E+01
F12	Avg	1.30114E+08	5.61811E+06	2.13577E+07	1.25391E+07
	Std	2.94786E+07	6.99101E+06	8.78266E+06	1.45163E+07
F13	Avg	2.55828E+07	1.51175E+04	3.22689E+04	1.53098E+03
	Std	7.02320E+07	5.89124E+03	2.18381E+04	1.63681E+02
F14	Avg	1.73265E+05	1.65988E+03	2.62817E+03	2.47539E+04
	Std	4.24911E+05	6.69880E+01	1.23753E+03	4.63129E+04
F15	Avg	2.24333E+04	4.69138E+03	1.98800E+04	7.35421E+03
	Std	1.23008E+04	2.59271E+03	1.88100E+04	5.66782E+03
F16	Avg	2.25472E+03	2.48335E+03	2.28480E+03	2.49303E+03
	Std	2.89748E+02	3.11469E+02	2.18389E+02	3.31359E+02
F17	Avg	1.93511E+03	2.13627E+03	2.04274E+03	1.99743E+03
	Std	1.05912E+02	1.62430E+02	1.31432E+02	9.41704E+01
F18	Avg	7.76360E+05	1.32882E+05	7.77843E+04	4.07748E+05
	Std	1.17899E+06	1.15102E+05	4.05328E+04	2.83313E+05
F19	Avg	7.49698E+04	1.15937E+04	1.09228E+04	4.84236E+03
	Std	1.47057E+05	9.37580E+03	7.92081E+03	3.56709E+03

Table 4 (continued)

Function	Criteria	GWO	SMA	MVO	FFA
F20	Avg	2.34981E+03	2.39439E+03	2.43296E+03	2.47696E+03
	Std	1.29281E+02	1.17279E+02	1.13577E+02	1.26477E+02
F21	Avg	2.37107E+03	2.22532E+03	2.17839E+03	2.18173E+03
	Std	1.53494E+02	2.18458E+01	3.52116E+01	**2.05381E+01**
F22	Avg	2.29353E+03	2.30995E+03	2.28760E+03	**2.28239E+03**
	Std	1.84437E+01	3.77420E+01	1.87930E+01	4.53255E+01
F23	Avg	2.87745E+03	**2.50000E+03**	2.87883E+03	2.83911E+03
	Std	5.36707E+01	**0.00000E+00**	3.25759E+01	1.42918E+01
F24	Avg	3.13805E+03	**2.60000E+03**	3.39295E+03	3.15728E+03
	Std	3.38016E+02	**0.00000E+00**	1.51690E+02	3.07222E+02
F25	Avg	3.16306E+03	**2.70000E+03**	2.92670E+03	2.93604E+03
	Std	1.41672E+02	**0.00000E+00**	3.01147E+01	3.40012E+01
F26	Avg	5.05476E+03	**2.80000E+03**	4.99376E+03	4.86555E+03
	Std	9.22154E+02	**0.00000E+00**	9.16611E+02	4.84829E+02
F27	Avg	3.72779E+03	**2.90000E+03**	3.59402E+03	3.45213E+03
	Std	1.31374E+02	**0.00000E+00**	8.93946E+01	2.17151E+01
F28	Avg	3.66284E+03	**3.00000E+03**	3.42376E+03	3.27594E+03
	Std	3.02204E+02	**0.00000E+00**	5.96603E+02	3.11772E+01
F29	Avg	3.58856E+03	**3.10000E+03**	3.72116E+03	3.48616E+03
	Std	1.84348E+02	**0.00000E+00**	1.25960E+02	1.90231E+02
F30	Avg	4.79348E+05	1.70455E+04	1.06627E+06	9.84282E+03
	Std	4.54157E+05	2.02049E+04	9.93293E+05	5.06693E+03

drawn: the DMCS algorithm is different from the other nine comparison algorithms and significantly outperforms the other comparison algorithms. The DMCS algorithm is ranked first in the Friedman test with a value of 3.161, slightly smaller than the CLPSO algorithm with a value of 3.462. Figure 7 plots and shows the convergence curves of the methods while solving the benchmark functions from CEC 2017. The curve corresponding to the DMCS algorithm decreases quickly and is close to the optimal value in the early stage. The optimal can search for solutions with higher accuracy, which indicates that the DMCS algorithm increases the population diversity, prevents the algorithm from falling into local optima, and improves the convergence efficiency.

4 Experiment for Renal Pathology Image Segmentation

To test the effectiveness of the DMCS algorithm for maximum entropy multi-threshold image segmentation, this subsection used the DMCS algorithm and six other algorithms, which are CS [32], WOA [30], Bat Algorithm (BA) [76], PSO [25], Salp Swarm Algorithm (SSA) [77], MVO [66], Biogeography-based Learning Particle Swarm Optimization

(BLPSO) [78] and an Improved WOA (IWOA) [79], to conduct multi-threshold image segmentation comparison experiments on eight renal pathology images of LN. In any image processing task [80–83], the utilized metric should be chosen so carefully. Hence, we used three indicators of Peak Signal to Noise Ratio (PSNR) [84], Feature Similarity Index (FSIM) [85] and Structural Similarity Index (SSIM) [86] to access the effect of the experiment. For further analysis, the experiment results are subjected to the Friedman test. The eight renal pathology images were obtained from electronic medical records of LN patients in the first affiliated hospital of Wenzhou medical university, China. This study was approved by the Medical Ethics Committee of the hospital and following the Declaration of Helsinki.

4.1 Experiment Setup

In this experiment, the population size of the swarm intelligence algorithm is set to 20 and runs independently 20 times, with each iteration running 100 times. The algorithm's fitness function is 2D Rényi entropy, and the image size was set to 512×512. The thresholds of image segmentation were taken as 4, 5, and 6 for three independent experiments. All experiments were carried out on a desk computer with

Table 5 The results of Avg and Std analysis of conventional algorithms comparison experiment

F	DMCS	DE	HHO	WOA	GWO	SMA	MVO	FFA
F1	1	3	7	5	8	6	4	2
F2	1	6	3	7	8	5	2	4
F3	2	7	3	6	8	4	1	5
F4	1	4	6	7	8	5	3	2
F5	6	5	7	8	2	4	3	1
F6	6	1	7	8	3	5	4	2
F7	6	3	7	8	4	5	2	1
F8	6	4	7	8	3	5	1	2
F9	6	1	7	8	3	5	4	2
F10	1	6	5	7	2	4	3	8
F11	3	2	6	7	8	4	5	1
F12	1	2	6	8	7	3	5	4
F13	2	1	7	6	8	4	5	3
F14	2	1	6	8	7	3	4	5
F15	2	1	7	8	6	3	5	4
F16	2	1	7	8	3	5	4	6
F17	6	2	7	8	1	5	4	3
F18	1	4	7	8	6	3	2	5
F19	1	3	7	8	6	5	4	2
F20	4	1	8	7	2	3	5	6
F21	1	4	6	7	8	5	2	3
F22	6	5	7	8	3	4	2	1
F23	1	5	1	8	6	1	7	4
F24	1	8	1	4	5	1	7	6
F25	1	5	1	1	8	1	6	7
F26	1	8	1	4	7	1	6	5
F27	1	4	1	8	7	1	6	5
F28	1	8	1	4	7	1	6	5
F29	1	4	1	8	6	1	7	5
F30	1	5	1	8	6	4	7	3
+/-/=	~	17/12/1	22/0/8	28/0/2	20/7/3	13/5/12	19/8/3	22/8/0
mean	2.50	3.80	4.93	6.93	5.53	3.53	4.20	3.73
rank	1	4	6	8	7	2	5	3
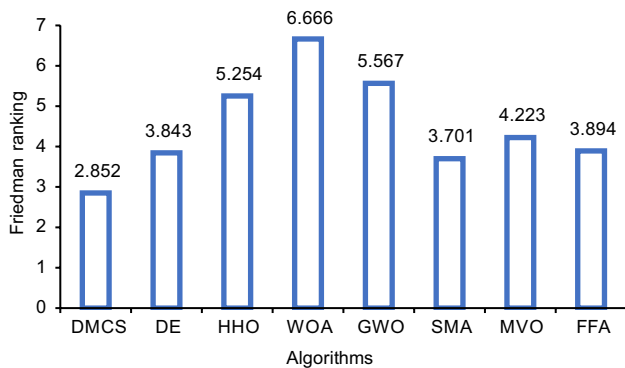
Fig. 4 The Freidman test result of DMCS and other conventional algorithms

Windows 10, Intel Core i5, 2.90 GHz, 8 GB of RAM, and MATLAB R2021b.

4.2 Multi-threshold Image Segmentation

Image segmentation is an essential technique in image processing that determines the quality of object detection, recognition, and tracking. The choice of image segmentation method directly influences the performance of an image analysis system. Thresholding has become a fundamental technique for image segmentation because of its stable performance and simplicity. The threshold segmentation method can be classified into two categories: single-threshold segmentation and multi-threshold segmentation. Multi-thresholds are used to set the multi-threshold to the image's grayscale, then compare each pixel's grayscale with the

Fig. 5 The convergence curves of DMCS and the other seven conventional algorithms

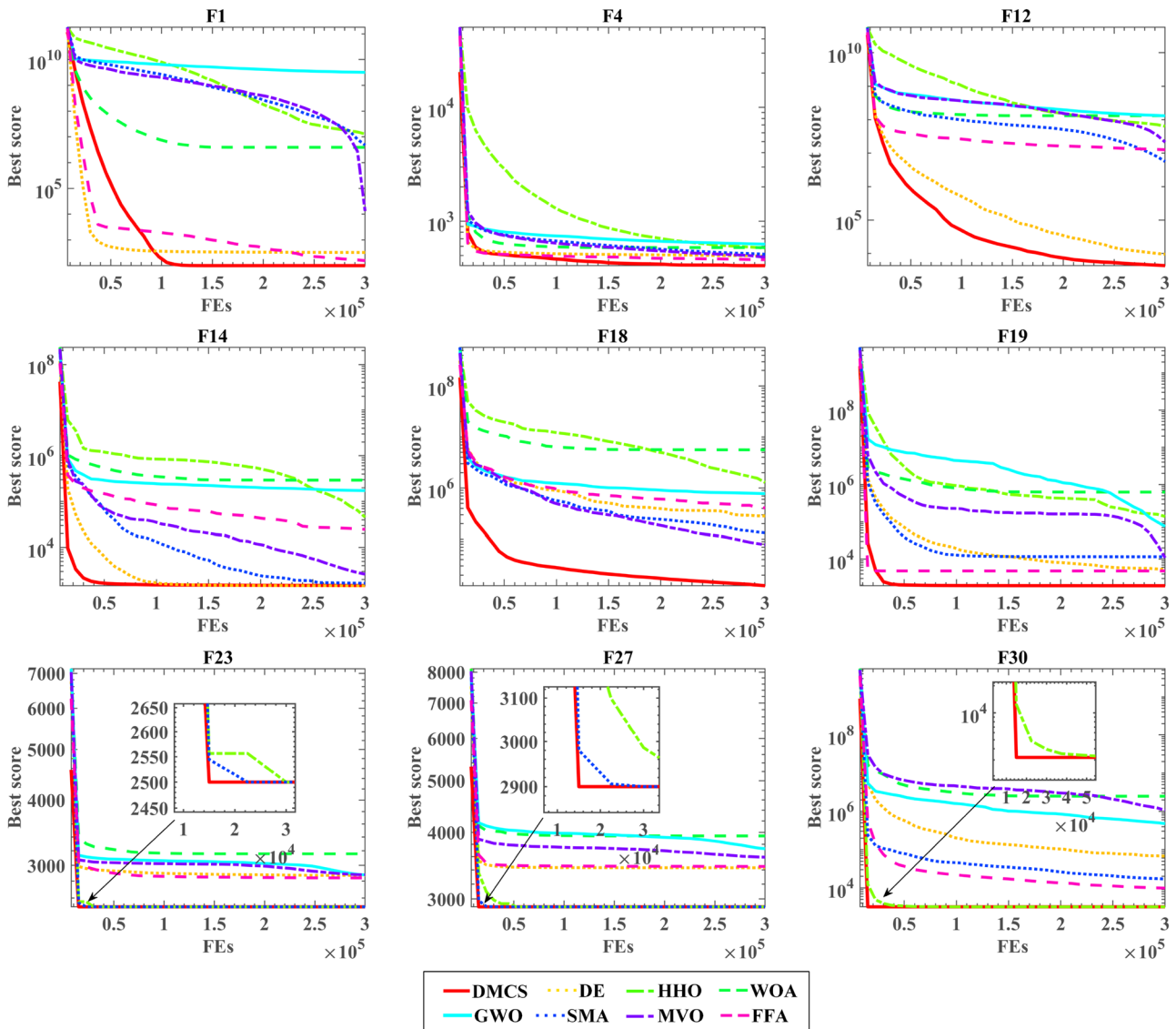threshold, and then classify it, which is a pattern recognition classification process [87, 88]. The traditional method has high computational intensity and inaccurate results [89]. The current swarm intelligence algorithm searches for the best threshold for simple calculation. This work uses the DMCS algorithm to search for the optimal threshold and compare it with six other swarm intelligence algorithms.

4.2.1 Non-Local Means (NLM) for 2D Histogram

French scholars Buades et al. [90] proposed an image denoising algorithm, the Non-Local Means (NLM) denoising algorithm, in 2005. Although image denoising and image segmentation method are two independent stages, the quality of image denoising can significantly impact image

segmentation. The original threshold segmentation method uses a 1D grayscale histogram to select the threshold, but it cannot segment the image containing noise. A 2D histogram-based multi-threshold image segmentation combining local pixel grayscale averages and original grayscale histograms proposed by Abutaleb et al. [91] solves this problem. However, using a 2D histogram to find the optimal threshold has high computational complexity, and the traditional 2D histogram cannot make the best of the edge and noise information of the image, resulting in inaccurate image segmentation [92, 93]. Hence, in this paper, the NLM image and grayscale image are used to generate a 2D histogram and then combined with the proposed DMCS algorithm to find the optimal threshold for medical image segmentation

Table 6 The results achieved by DMCS against the nine advanced algorithms

Algorithms	Criteria	F1	F2	F3	F4	F5	F6
DMCS	Avg	**1.00000E+02**	**4.41733E+03**	**3.03069E+02**	**4.09392E+02**	6.41217E+02	6.30844E+02
	Std	**1.20311E−11**	**6.82898E+03**	**3.21576E+00**	2.26560E+01	2.82708E+01	5.80851E+00
OBLGWO	Avg	1.30457E+07	5.34738E+18	6.00814E+03	5.55367E+02	6.55682E+02	6.16384E+02
	Std	8.32219E+06	1.43801E+19	2.22980E+03	4.05906E+01	3.87480E+01	1.39971E+01
MSCA	Avg	1.03118E+04	1.43637E+213	8.34367E+04	4.63464E+02	6.83208E+02	6.03342E+02
	Std	1.49952E+04	6.55350E+04	2.28821E+04	3.88712E+01	3.39804E+01	3.27787E+00
HGWO	Avg	1.06035E+10	6.19168E+31	7.47658E+04	9.32041E+02	7.04517E+02	6.31034E+02
	Std	2.26331E+09	2.92212E+32	6.07490E+03	6.82724E+01	1.22247E+01	2.79272E+00
CLPSO	Avg	1.04707E+02	7.90890E+13	8.60573E+03	4.70231E+02	**5.53534E+02**	**6.00000E+02**
	Std	1.90913E+01	2.68563E+14	1.63701E+03	**1.84441E+01**	**7.22968E+00**	**7.00176E−14**
ALCPSO	Avg	1.51919E+03	1.77931E+18	4.29753E+03	5.18606E+02	5.81074E+02	6.00842E+02
	Std	1.66951E+03	5.50512E+18	5.94834E+02	5.08663E+01	1.66525E+01	6.99634E−01
LWOA	Avg	5.27896E+05	7.53188E+05	3.25500E+02	5.14744E+02	7.06369E+02	6.56585E+02
	Std	1.76822E+05	1.14934E+06	7.31263E+00	3.54362E+01	4.02097E+01	8.06200E+00
RDWOA	Avg	1.29963E+05	4.77189E+04	3.22844E+02	4.94821E+02	6.49670E+02	6.00587E+02
	Std	3.80573E+04	1.05945E+05	2.05347E+01	3.36547E+01	4.37085E+01	6.59443E−01
OMGSCA	Avg	3.18927E+09	1.32486E+28	1.73880E+04	6.38976E+02	6.03256E+02	6.02203E+02
	Std	2.22120E+09	4.57860E+28	4.18772E+03	5.17755E+01	1.65846E+01	2.54311E+00
GBHHO	Avg	2.45662E+03	4.40345E+08	3.74619E+02	4.95648E+02	6.43842E+02	6.05566E+02
	Std	2.70898E+03	2.19582E+09	4.97346E+01	4.04359E+01	2.50958E+01	4.45746E+00

Algorithms	Criteria	F7	F8	F9	F10	F11	F12
DMCS	Avg	1.03983E+03	9.63471E+02	4.26049E+03	3.80984E+03	1.22915E+03	**3.73222E+03**
	Std	8.34478E+01	4.62963E+01	6.50324E+02	3.31771E+02	5.86921E+01	**8.95992E+02**
OBLGWO	Avg	9.41907E+02	9.48419E+02	4.64030E+03	4.62466E+03	1.44261E+03	8.38527E+07
	Std	7.61738E+01	4.14696E+01	2.61696E+03	7.55604E+02	8.55487E+01	5.19384E+07
MSCA	Avg	9.93177E+02	9.96105E+02	6.06473E+03	3.85537E+03	1.47947E+03	8.56595E+06
	Std	7.18638E+01	3.06302E+01	2.12075E+03	3.55971E+02	3.03610E+02	2.07386E+07
HGWO	Avg	1.06643E+03	1.02173E+03	4.84461E+03	6.73498E+03	7.47037E+03	8.11053E+08
	Std	3.45549E+01	1.59510E+01	5.83338E+02	3.07947E+02	1.64392E+03	2.43333E+08
CLPSO	Avg	**7.84862E+02**	**8.48600E+02**	**9.21539E+02**	**2.98690E+03**	**1.14272E+03**	4.22407E+05
	Std	**1.03947E+01**	**6.46136E+00**	**1.92989E+01**	**2.93461E+02**	**6.79116E+00**	7.32666E+05
ALCPSO	Avg	8.46268E+02	8.76085E+02	1.28422E+03	3.19898E+03	1.18876E+03	8.09262E+03
	Std	3.19635E+01	2.51483E+01	5.09257E+02	4.68037E+02	4.78952E+01	5.46687E+03
LWOA	Avg	1.15299E+03	1.07616E+03	7.99264E+03	5.12229E+03	1.35847E+03	2.48953E+07
	Std	9.06091E+01	4.29968E+01	2.89651E+03	4.92608E+02	8.65387E+01	1.16452E+07
RDWOA	Avg	9.43890E+02	9.56176E+02	5.17412E+03	4.11158E+03	1.24549E+03	6.18504E+05
	Std	5.47418E+01	3.17080E+01	1.68378E+03	6.41964E+02	5.49793E+01	4.17048E+05
OMGSCA	Avg	9.20609E+02	9.24748E+02	2.91907E+03	3.30879E+03	1.47878E+03	7.40081E+07
	Std	6.37010E+01	2.22658E+01	5.79527E+02	3.78182E+02	9.71917E+01	3.07081E+07
GBHHO	Avg	9.43111E+02	9.58947E+02	4.36070E+03	4.06969E+03	1.22478E+03	1.72155E+06
	Std	4.98769E+01	3.14084E+01	9.75344E+02	4.86380E+02	4.51639E+01	3.07198E+06

Algorithms	Criteria	F13	F14	F15	F16	F17	F18
DMCS	Avg	**1.39883E+03**	**1.46680E+03**	**1.59886E+03**	2.29362E+03	2.13142E+03	**1.46452E+04**
	Std	**2.47371E+01**	**3.46753E+01**	**4.43161E+01**	2.54910E+02	1.61947E+02	**8.83622E+03**
OBLGWO	Avg	2.94959E+05	2.83395E+04	5.06428E+04	2.61060E+03	2.18307E+03	5.26607E+05
	Std	1.81344E+05	3.18799E+04	4.12959E+04	3.14893E+02	1.75226E+02	4.73149E+05
MSCA	Avg	9.53594E+03	1.91813E+05	3.15309E+04	2.84589E+03	2.43775E+03	3.00674E+05
	Std	1.32423E+04	1.57034E+05	5.47788E+04	3.72391E+02	2.06421E+02	3.40727E+05

Table 6 (continued)

Algorithms	Criteria	F13	F14	F15	F16	F17	F18
HGWO	Avg	9.22316E+07	7.24557E+05	9.53450E+06	3.05818E+03	2.26661E+03	1.28187E+06
	Std	3.69779E+07	4.20065E+05	9.25231E+06	2.12137E+02	1.37505E+02	9.11788E+05
CLPSO	Avg	1.41234E+03	3.88702E+03	1.60917E+03	**2.03346E+03**	**1.90814E+03**	1.13805E+05
	Std	6.84425E+01	3.07777E+03	8.00351E+01	**1.10113E+02**	**4.72514E+01**	7.94612E+04
ALCPSO	Avg	1.49260E+03	1.46630E+03	1.61080E+03	2.23574E+03	1.99048E+03	5.04827E+04
	Std	1.16676E+02	1.88408E+01	2.43581E+01	2.78274E+02	1.12783E+02	2.80576E+04
LWOA	Avg	7.30842E+04	6.51417E+03	5.02301E+04	2.80207E+03	2.40898E+03	3.66096E+05
	Std	5.82316E+04	5.51857E+03	5.76599E+04	3.23472E+02	2.10966E+02	3.97534E+05
RDWOA	Avg	3.39226E+03	3.46439E+03	3.80363E+03	2.53137E+03	2.06050E+03	1.72771E+05
	Std	2.47225E+03	1.53363E+03	2.18498E+03	3.02699E+02	1.09415E+02	1.45200E+05
OMGSCA	Avg	5.75234E+03	5.71585E+03	3.20934E+03	2.21221E+03	1.99233E+03	3.44858E+05
	Std	2.75458E+03	7.81275E+03	9.77247E+02	1.95816E+02	1.05669E+02	4.05656E+05
GBHHO	Avg	5.30809E+03	1.56964E+03	4.56252E+03	2.61863E+03	2.16471E+03	1.83982E+05
	Std	3.16417E+03	8.58666E+01	4.51342E+03	3.14721E+02	1.87452E+02	1.46482E+05

Algorithms	Criteria	F19	F20	F21	F22	F23	F24
DMCS	Avg	**1.93202E+03**	2.39297E+03	**2.12321E+03**	2.38219E+03	**2.50000E+03**	**2.60000E+03**
	Std	**8.14186E+00**	9.48592E+01	3.69290E+01	3.74778E+01	**0.00000E+00**	**0.00000E+00**
OBLGWO	Avg	1.99797E+05	2.52833E+03	2.24113E+03	2.35859E+03	2.95357E+03	2.65992E+03
	Std	2.00599E+05	1.73193E+02	3.34331E+01	4.20539E+01	5.27019E+01	2.28017E+02
MSCA	Avg	1.13286E+04	2.56542E+03	2.17691E+03	2.40509E+03	2.94061E+03	3.00019E+03
	Std	1.08667E+04	1.67099E+02	3.66456E+01	4.34116E+01	8.62486E+01	1.48324E+02
HGWO	Avg	7.44612E+06	2.67314E+03	2.64922E+03	2.43041E+03	3.03563E+03	3.07412E+03
	Std	2.76051E+06	1.60784E+02	8.56082E+01	1.49642E+01	5.56900E+01	3.70231E+02
CLPSO	Avg	1.97217E+03	**2.21179E+03**	2.17873E+03	**2.25315E+03**	2.84375E+03	2.61605E+03
	Std	7.11005E+01	**5.85629E+01**	**1.65819E+01**	**6.32211E+00**	1.14179E+01	1.95285E+01
ALCPSO	Avg	7.75836E+03	2.22078E+03	2.21411E+03	2.28251E+03	2.97078E+03	3.04522E+03
	Std	7.73559E+03	7.88318E+01	3.48740E+01	2.34630E+01	8.41748E+01	4.31701E+02
LWOA	Avg	9.26139E+04	2.65588E+03	2.22647E+03	2.43691E+03	3.12991E+03	2.76606E+03
	Std	7.70317E+04	1.74584E+02	3.39448E+01	4.57195E+01	1.19208E+02	3.73288E+02
RDWOA	Avg	6.48689E+03	2.34235E+03	2.20439E+03	2.37204E+03	2.50000E+03	2.60000E+03
	Std	4.91624E+03	1.29495E+02	3.44768E+01	3.56431E+01	7.52734E−11	6.10549E−11
OMGSCA	Avg	2.11849E+03	2.31660E+03	2.34004E+03	2.32518E+03	2.93951E+03	2.60004E+03
	Std	1.98097E+02	9.94699E+01	5.93197E+01	2.30564E+01	3.76358E+01	5.79615E−02
GBHHO	Avg	5.34532E+03	2.45553E+03	2.21398E+03	2.35553E+03	**2.50000E+03**	**2.60000E+03**
	Std	4.34913E+03	1.54172E+02	2.79199E+01	2.74419E+01	**0.00000E+00**	**0.00000E+00**

Algorithms	Criteria	F25	F26	F27	F28	F29	F30
DMCS	Avg	**2.70000E+03**	**2.80000E+03**	**2.90000E+03**	**3.00000E+03**	**3.10000E+03**	**3.20000E+03**
	Std	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
OBLGWO	Avg	**2.70000E+03**	3.82441E+03	3.64338E+03	3.22774E+03	3.11837E+03	4.07592E+05
	Std	**0.00000E+00**	1.60616E+03	9.35739E+01	4.08427E+02	7.23581E+01	4.46650E+05
MSCA	Avg	2.95941E+03	5.72288E+03	3.20001E+03	3.29547E+03	3.98372E+03	9.26878E+03
	Std	1.17945E+02	8.19079E+02	3.01200E−04	1.58508E+01	2.88159E+02	1.32173E+04
HGWO	Avg	3.38814E+03	3.81843E+03	3.97145E+03	3.78484E+03	3.14175E+03	3.20000E+03
	Std	1.45386E+02	1.61744E+03	3.04478E+02	1.39536E+02	9.27542E+01	1.59465E−08
CLPSO	Avg	2.90845E+03	3.68824E+03	3.50831E+03	3.28135E+03	3.38234E+03	1.46691E+04
	Std	1.19940E+01	6.10220E+02	2.85086E+01	1.38746E+01	6.72333E+01	7.88961E+03
ALCPSO	Avg	2.96339E+03	5.25755E+03	3.75556E+03	3.48495E+03	3.48150E+03	4.73729E+04
	Std	5.38191E+01	8.58031E+02	1.29768E+02	5.95141E+02	1.29532E+02	4.95066E+04

Table 6 (continued)

Algorithms	Criteria	F25	F26	F27	F28	F29	F30
LWOA	Avg	2.77966E+03	3.64752E+03	3.88787E+03	3.19170E+03	3.94497E+03	1.04126E+06
	Std	1.19372E+02	1.70421E+03	1.29924E+02	1.32307E+02	2.51911E+02	4.36578E+05
RDWOA	Avg	2.70000E+03	2.80000E+03	2.95660E+03	3.00000E+03	3.10612E+03	2.66636E+04
	Std	4.95839E−10	9.33973E−10	2.16564E+02	1.62614E−10	3.35032E+01	1.44598E+04
OMGSCA	Avg	2.75068E+03	2.80035E+03	3.59543E+03	3.48365E+03	3.58403E+03	2.81389E+05
	Std	1.47979E+02	4.94447E−01	7.89469E+01	1.33162E+02	1.28402E+02	3.07717E+05
GBHHO	Avg	**2.70000E+03**	**2.80000E+03**	**2.90000E+03**	**3.00000E+03**	**3.10000E+03**	1.76300E+04
	Std	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	2.39865E+04

Table 7 The results of Avg and Std analysis of advanced algorithms comparison experiment

| F	DMCS	OBLGWO	MSCA	HGWO	CLPSO	ALCPSO	LWOA	RDWOA	OMGSCA	GBHHO
F1	1	8	5	10	2	3	7	6	9	4
F2	2	8	1	10	6	7	4	3	9	5
F3	1	6	10	9	7	5	3	2	8	4
F4	1	8	2	10	3	7	6	4	9	5
F5	4	7	8	9	1	2	10	6	3	5
F6	8	7	5	9	1	3	10	2	4	6
F7	8	4	7	9	1	2	10	6	3	5
F8	7	4	8	9	1	2	10	5	3	6
F9	4	6	9	7	1	2	10	8	3	5
F10	4	8	5	10	1	2	9	7	3	6
F11	4	7	9	10	1	2	6	5	8	3
F12	1	9	6	10	3	2	7	4	8	5
F13	1	9	7	10	2	3	8	4	6	5
F14	2	8	9	10	5	1	7	4	6	3
F15	1	9	7	10	2	3	8	5	4	6
F16	4	6	9	10	1	3	8	5	2	7
F17	5	7	10	8	1	2	9	4	3	6
F18	1	9	6	10	3	2	8	4	7	5
F19	1	9	7	10	2	6	8	5	3	4
F20	5	7	8	10	1	2	9	4	3	6
F21	1	8	2	10	3	6	7	4	9	5
F22	7	5	8	9	1	2	10	6	3	4
F23	1	7	6	9	4	8	10	3	5	1
F24	1	6	8	10	5	9	7	3	4	1
F25	1	1	8	10	7	9	6	4	5	1
F26	1	8	10	7	6	9	5	3	4	1
F27	1	7	4	10	5	8	9	3	6	1
F28	1	5	7	10	6	9	4	3	8	1
F29	1	4	10	5	6	7	9	3	8	1
F30	1	9	3	2	4	7	10	6	8	5
+/−/=	~	20/2/8	26/2/2	28/0/2	17/11/2	17/10/3	30/0/0	16/3/11	20/9/1	14/3/13
mean	2.70	6.87	6.80	9.07	3.07	4.50	7.80	4.37	5.47	4.07
rank	1	8	7	10	2	5	9	4	6	3
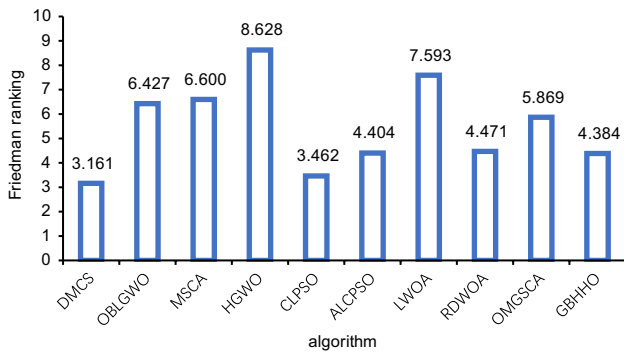
Fig. 6 The Friedman test result of DMCS and other advanced algorithms

purposes while reducing the to reduce the computational complexity.

NLM denoising algorithm calculates the weighted sum of the pixel values of all pixels in the rectangular window, and the weights obey the Gaussian distribution. It uses the similarity between the neighborhood block of the current filter point and the neighborhood blocks of other points in the rectangular window to calculate the weight. The greater the similarity, the greater the weight. A simple formula for NLM will be given below:
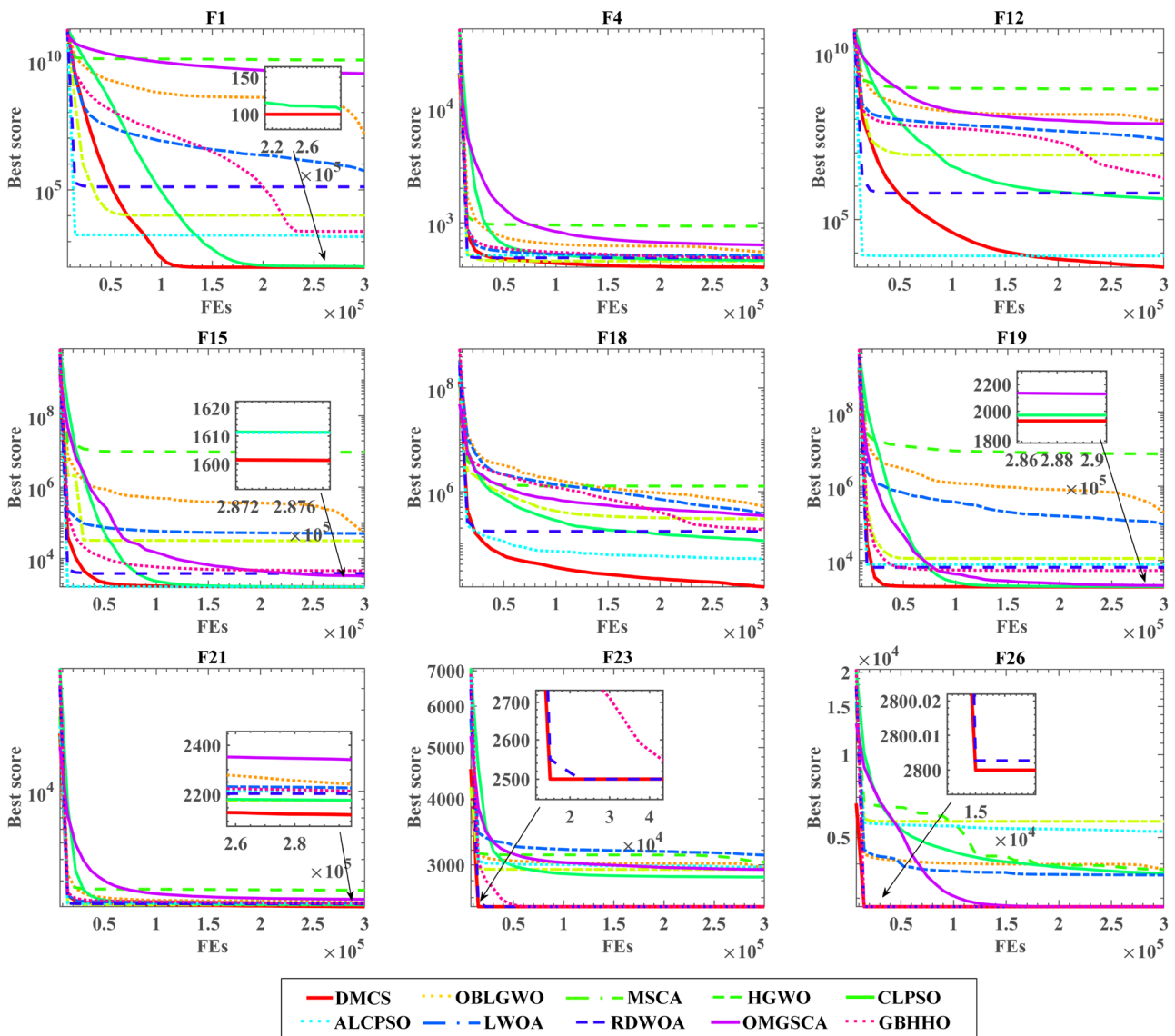

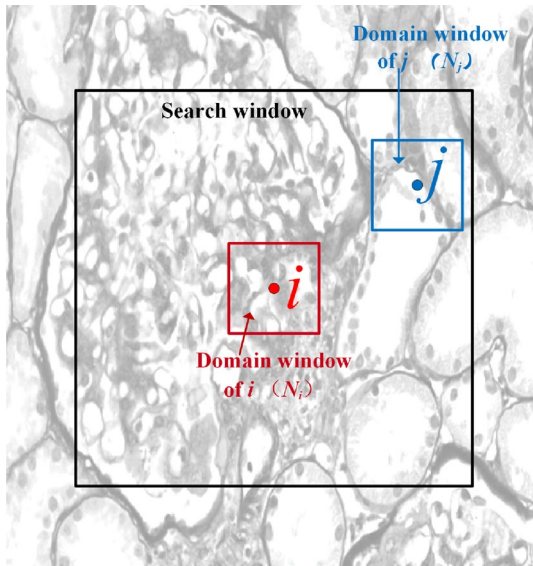**Fig. 7** The convergence curves of DMCS and other nine advanced algorithms
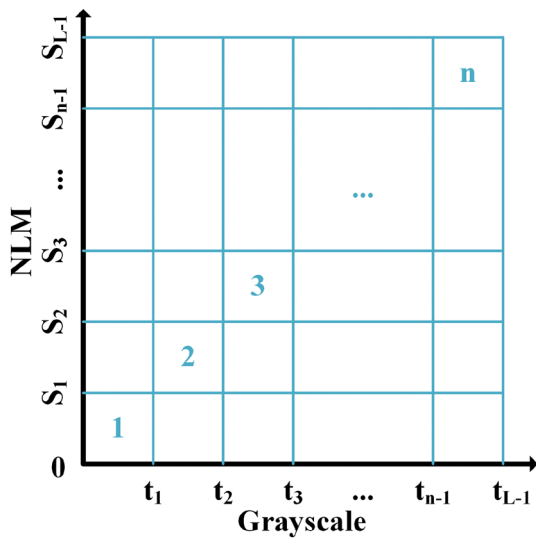
Fig. 8 The schematic diagram of NLM

Fig. 9 NLM 2D histogram

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j), \tag{11}$$

$$w(i,j) = \frac{1}{Z(i)} e^{\frac{-\|v(N_i)-v(N_j)\|_{2,a}^2}{h^2}}, \tag{12}$$

$$Z(i) = \sum_{j} e^{\frac{-\|v(N_i)-v(N_j)\|_{2,a}^2}{h^2}}. \tag{13}$$

The formulas will be explained below in conjunction with Fig. 8. In the figure, the window with a black border is the search window centered on the target pixel i. The small red and blue windows are the domain windows centered on pixels i and j severally. $v = \{v(i)|i \in I\}$ denotes a discrete noisy image. $NL[v](i)$ is the filter value of point $v(i)$, it is obtained from the weighted average of the pixel values of all points in the search window. $w(i,j)$ denotes the Gaussian weight between $v(i)$ and any point $v(j)$ in the search window, calculated from the mean-squared error $v(N_i) - v(N_j)_{2,a}^2$ of the respective domain block of the two points, and $w(i,j)$ lies between 0 and 1, and $\sum_j w(i,j) = 1$. $v(N_i)$ and $v(N_j)$ are the intensity gray level vectors, where N_k denotes a square window centered at pixel k. The more similar $v(N_i)$ and $v(N_j)$ are, the larger $w(i,j)$ is. The similarity between pixel i and pixel j is determined by the similarity of $w(i,j)$. $a > 0$ is the standard deviation of the Gaussian kernel. $Z(i)$ is a normalizing factor. h is a coefficient to control the decay of the exponential function. The size of the h-value is proportional to the denoising effect and inversely proportional to the degree of image distortion. In other words, the larger the h, the better the denoising effect, but the more blurred the image will be. The smaller the h, the worse the denoising effect, but the smaller the denoising distortion.

Suppose an image I with a total number of $m * n$ pixels is divided into L gray levels, then the gray value range of pixels in the original image I is from 0 to $L - 1$, and the gray value range of pixels in the NLM image is from 0 to $L - 1$, too. The grayscale probability density of pixels in the NLM for 2D histogram generated by combining the grayscale image and the NLM image is calculated by Eq. (14) [94].

$$p(i,j) = \frac{f(i,j)}{m \times n}, \tag{14}$$

where the frequency of occurrence of the binary group (i,j) is denoted by $f(i,j)$. i denotes the pixel gray level of the original image I, j denotes the pixel gray level of the NLM image, and Fig. 9 shows the NLM 2D histogram, where the pixel values of the original grayscale image I are represented by the horizontal axis and the NLM image are represented by the vertical axis, with the axis intervals determined by the corresponding threshold levels of the image.

4.2.2 2D Rényi Entropy Mode

2D Rényi entropy is a generalization of Shannon entropy, which differs from Shannon entropy in those variable parameters α are introduced into it. Then in 2008, Leila et al. defined the conditional 2D Rényi entropy and showed that the so-called chain rule holds for 2D Rényi entropy [95–97].
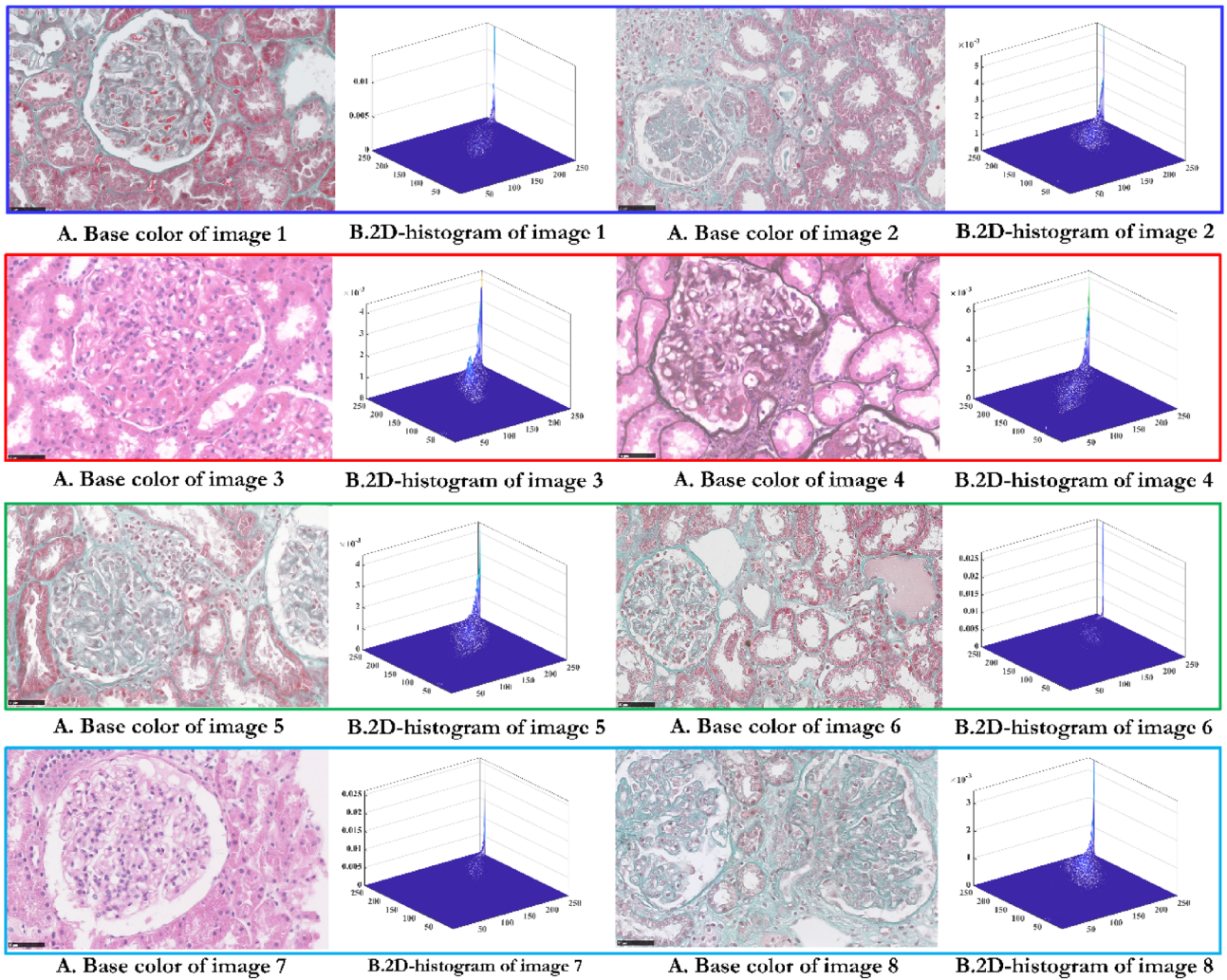
image 1: Glomerular micropathy (trichrome stained);
image 2: LN class IV (trichrome stained);
image 3: LN class III+V (Haematoxylin and Eosin stained);
image 4: LN class III+V (periodic acid-siliver metheramine and H&E stained);
image 5: treated LN with glomerular micropathological changes, glomerular sclerosis and fibrous crescent formation (trichrome stained);
image 6: LN class I (trichrome stained);
image 7: treated LN with glomerular sclerosis (Periodic Acid Schiff stained);
image 8: LN class IV+V (trichrome stained);

Fig. 10 Original images and corresponding 2D histograms of the eight images used in the experiment

In this paper, the coefficient of α is 0.5. Here, we introduce the 2D Rényi entropy method of multi-threshold.

Given an image I divided into L gray levels, the distribution probability of each gray level is $\{p(1), p(2), \dots, p(i), \dots, p(L)\}$. Assuming that there are N thresholds $\{t_1, t_2, \dots, t_N\}$, the histogram is divided into $N + 1$ regions, and the grayscale probability of the first threshold t can be described as Eq. (15):

$$w_j(t) = \begin{cases} \sum_{i=1}^{t_1} p(i), j = 1 \\ \sum_{i=t_{n-1}+1}^{t_n} p(i), 1 < j < N \\ \sum_{i=t_{N-1}+1}^{L-1} p(i), \text{otherelse} \end{cases} \quad (15)$$

The 2D Rényi entropy of each region Rt_α^n can be calculated by Eq. (16):

Table 8 The FSIM of each algorithm

Image	K		DMCS	CS	WOA	BA	PSO
1	4	Avg	**9.12095E–01**	8.55712E–01	8.94355E–01	9.02780E–01	9.04492E–01
		Std	**3.23578E–02**	6.16052E–02	5.76997E–02	4.55442E–02	4.94625E–02
	5	Avg	**9.41737E–01**	8.85851E–01	9.05817E–01	8.97601E–01	9.09349E–01
		Std	**1.58398E–02**	4.75213E–02	6.85609E–02	1.15582E–01	6.05325E–02
	6	Avg	**9.67603E–01**	9.15102E–01	9.25680E–01	9.26205E–01	9.23820E–01
		Std	**8.71950E–03**	2.99607E–02	4.42501E–02	7.09853E–02	6.06237E–02
2	4	Avg	**9.32471E–01**	9.12391E–01	8.63701E–01	8.81901E–01	9.00947E–01
		Std	**1.13890E–02**	3.55858E–02	1.28538E–01	4.61051E–02	8.89203E–02
	5	Avg	**9.66917E–01**	9.17143E–01	9.04833E–01	8.98700E–01	9.44370E–01
		Std	1.39479E–02	5.67016E–02	5.40122E–02	5.58035E–02	2.48847E–02
	6	Avg	**9.72422E–01**	9.40835E–01	9.45702E–01	9.15370E–01	9.38046E–01
		Std	**1.51980E–02**	3.50007E–02	3.95189E–02	5.42792E–02	5.29058E–02
3	4	Avg	**8.42834E–01**	8.26255E–01	7.93962E–01	7.63694E–01	8.29403E–01
		Std	**2.59818E–02**	5.54108E–02	5.66070E–02	7.52358E–02	3.91362E–02
	5	Avg	**8.75410E–01**	8.48521E–01	8.54091E–01	8.09504E–01	8.61721E–01
		Std	**2.20396E–02**	6.71862E–02	3.66625E–02	8.49594E–02	3.58040E–02
	6	Avg	**9.04271E–01**	8.99670E–01	8.69808E–01	8.46003E–01	8.79478E–01
		Std	**3.14729E–02**	3.81930E–02	6.33001E–02	6.34398E–02	4.02192E–02
4	4	Avg	**8.70921E–01**	8.30477E–01	8.46213E–01	8.37307E–01	8.47302E–01
		Std	**1.88736E–02**	5.64860E–02	3.99347E–02	4.71795E–02	3.60582E–02
	5	Avg	8.95719E–01	**8.96853E–01**	8.66150E–01	8.51004E–01	8.86392E–01
		Std	2.65975E–02	3.67655E–02	5.59203E–02	5.49673E–02	3.97594E–02
	6	Avg	9.05393E–01	8.99389E–01	**9.12684E–01**	9.00351E–01	9.12350E–01
		Std	**2.51329E–02**	3.47847E–02	2.90664E–02	3.38040E–02	2.88702E–02
5	4	Avg	**8.89430E–01**	8.66679E–01	8.60772E–01	8.29551E–01	8.67201E–01
		Std	**1.88808E–02**	3.86681E–02	5.24221E–02	1.35976E–01	4.34719E–02
	5	Avg	**9.16675E–01**	8.98453E–01	8.66662E–01	8.62539E–01	8.90484E–01
		Std	**2.08842E–02**	5.76518E–02	1.39611E–01	1.00058E–01	4.02707E–02
	6	Avg	9.24528E–01	9.10929E–01	9.25919E–01	8.86999E–01	**9.28271E–01**
		Std	2.11194E–02	4.20662E–02	**1.88382E–02**	7.27049E–02	2.14119E–02
6	4	Avg	**9.42141E–01**	9.25780E–01	9.17638E–01	9.25043E–01	9.36581E–01
		Std	**1.65329E–02**	2.87857E–02	3.63279E–02	3.17377E–02	2.34694E–02
	5	Avg	**9.57359E–01**	9.15217E–01	9.36268E–01	9.25662E–01	9.49227E–01
		Std	**1.32212E–02**	5.00399E–02	3.46303E–02	3.93797E–02	3.07925E–02
	6	Avg	**9.68759E–01**	9.35337E–01	9.29263E–01	9.20071E–01	9.57371E–01
		Std	**7.56913E–03**	3.04785E–02	1.06075E–01	1.09528E–01	2.47703E–02
7	4	Avg	**9.09565E–01**	8.80857E–01	8.78068E–01	8.96670E–01	8.94152E–01
		Std	**2.02225E–02**	4.73120E–02	5.21113E–02	3.04699E–02	2.90075E–02
	5	Avg	**9.28638E–01**	9.11989E–01	9.16099E–01	9.07335E–01	9.17289E–01
		Std	**1.74842E–02**	3.64391E–02	3.48813E–02	4.50679E–02	3.14956E–02
	6	Avg	**9.49408E–01**	9.28715E–01	9.40424E–01	9.21459E–01	9.31816E–01
		Std	**1.77207E–02**	2.64026E–02	2.65673E–02	5.68528E–02	2.47655E–02
8	4	Avg	**9.24065E–01**	8.85474E–01	8.73538E–01	8.37485E–01	8.95073E–01
		Std	**2.70344E–02**	5.91464E–02	6.24384E–02	7.89888E–02	3.74878E–02
	5	Avg	**9.56184E–01**	9.25488E–01	9.27445E–01	8.51160E–01	9.27209E–01
		Std	**1.42652E–02**	3.40218E–02	3.46696E–02	1.65287E–01	3.42710E–02
	6	Avg	**9.69208E–01**	9.34667E–01	9.32894E–01	9.33587E–01	9.51835E–01
		Std	**1.18349E–02**	5.41263E–02	4.24641E–02	3.63077E–02	2.44825E–02

Image	K		SSA	MVO	BLPSO	IWOA
1	4	Avg	8.92739E–01	8.43078E–01	8.31110E–01	8.87466E–01

Table 8 (continued)

Image	K			SSA	MVO	BLPSO	IWOA
			Std	4.66734E–02	6.04366E–02	6.14806E–02	5.10184E–02
	5		Avg	9.07265E–01	8.80311E–01	8.54390E–01	9.13003E–01
			Std	5.71636E–02	5.45244E–02	5.56354E–02	3.82858E–02
	6		Avg	9.14770E–01	8.63819E–01	8.80559E–01	8.91461E–01
			Std	4.95044E–02	7.93804E–02	4.24161E–02	5.60901E–02
2	4		Avg	8.96794E–01	8.70735E–01	8.69786E–01	8.76432E–01
			Std	6.00016E–02	6.11635E–02	4.90666E–02	6.74629E–02
	5		Avg	9.27622E–01	8.98780E–01	8.56554E–01	9.06695E–01
			Std	3.89963E–02	5.76244E–02	**1.08392E–01**	6.62896E–02
	6		Avg	9.30011E–01	8.99838E–01	8.70705E–01	9.11145E–01
			Std	4.59686E–02	6.31111E–02	5.80955E–02	5.47448E–02
3	4		Avg	7.63596E–01	7.71614E–01	7.87113E–01	7.86774E–01
			Std	7.46362E–02	6.95541E–02	6.56654E–02	9.95578E–02
	5		Avg	8.71838E–01	8.14724E–01	7.63408E–01	8.46204E–01
			Std	3.56278E–02	9.31933E–02	7.01325E–02	4.20298E–02
	6		Avg	8.77081E–01	8.53703E–01	8.29528E–01	8.58250E–01
			Std	6.93123E–02	8.85052E–02	7.61705E–02	6.33720E–02
4	4		Avg	8.50759E–01	8.36848E–01	8.33470E–01	8.26085E–01
			Std	3.43742E–02	5.05237E–02	6.08436E–02	6.01389E–02
	5		Avg	8.96778E–01	8.31263E–01	8.51808E–01	8.54716E–01
			Std	**2.14465E–02**	7.00724E–02	5.47908E–02	4.11562E–02
	6		Avg	9.02983E–01	9.02993E–01	8.80108E–01	8.88989E–01
			Std	2.77117E–02	3.72693E–02	4.49210E–02	4.62821E–02
5	4		Avg	8.81447E–01	8.54893E–01	8.26870E–01	8.55273E–01
			Std	3.63539E–02	6.53466E–02	4.66205E–02	4.15452E–02
	5		Avg	8.84299E–01	8.74963E–01	8.34237E–01	8.98226E–01
			Std	4.42321E–02	5.12329E–02	5.19240E–02	3.08624E–02
	6		Avg	9.19614E–01	8.69529E–01	8.82270E–01	8.72273E–01
			Std	2.49629E–02	8.05603E–02	4.49473E–02	8.61659E–02
6	4		Avg	9.28453E–01	8.80849E–01	9.15626E–01	9.21653E–01
			Std	4.03064E–02	4.93754E–02	4.44837E–02	3.51225E–02
	5		Avg	9.35525E–01	8.91051E–01	9.25064E–01	9.31112E–01
			Std	4.10856E–02	4.87975E–02	3.39234E–02	2.81165E–02
	6		Avg	9.51754E–01	9.16032E–01	9.29669E–01	9.23265E–01
			Std	3.12011E–02	4.14442E–02	3.60871E–02	6.52778E–02
7	4		Avg	8.95961E–01	8.68320E–01	8.56121E–01	8.61611E–01
			Std	2.99361E–02	3.98598E–02	5.46917E–02	4.71546E–02
	5		Avg	9.12257E–01	8.95422E–01	8.60800E–01	8.89618E–01
			Std	5.28750E–02	5.38159E–02	5.67734E–02	5.39163E–02
	6		Avg	9.27454E–01	8.86419E–01	8.85829E–01	9.16703E–01
			Std	2.99597E–02	9.14963E–02	6.21634E–02	4.47252E–02
8	4		Avg	9.06003E–01	8.65700E–01	8.74101E–01	8.66739E–01
			Std	5.45133E–02	1.20668E–01	5.56094E–02	6.89638E–02
	5		Avg	9.08298E–01	8.57003E–01	8.69605E–01	8.74674E–01
			Std	5.27693E–02	7.31942E–02	5.86274E–02	7.40425E–02
	6		Avg	9.40570E–01	9.00725E–01	8.96484E–01	9.01420E–01
			Std	4.70278E–02	6.46651E–02	5.15822E–02	7.36046E–02

Table 9 The PSNR of each algorithm

Image	K		DMCS	CS	WOA	BA	PSO
1	4	Avg	**2.07778E+01**	1.82488E+01	1.93897E+01	1.97668E+01	1.95201E+01
		Std	**1.68543E+00**	2.90143E+00	2.88656E+00	2.22679E+00	2.92031E+00
	5	Avg	**2.29014E+01**	1.97002E+01	2.08672E+01	2.05798E+01	2.02511E+01
		Std	**6.59980E−01**	2.21714E+00	3.12410E+00	3.21877E+00	3.44437E+00
	6	Avg	**2.44850E+01**	2.15858E+01	2.16607E+01	2.20024E+01	2.16430E+01
		Std	**8.27912E−01**	1.50477E+00	2.57220E+00	3.16508E+00	3.03666E+00
2	4	Avg	**2.21856E+01**	2.09805E+01	1.98613E+01	1.92243E+01	2.08829E+01
		Std	**5.39345E−01**	1.61592E+00	3.34480E+00	2.53997E+00	2.96586E+00
	5	Avg	**2.43295E+01**	2.20707E+01	2.12441E+01	2.06177E+01	2.30091E+01
		Std	**8.01944E−01**	1.64923E+00	3.12786E+00	2.63540E+00	1.38594E+00
	6	Avg	**2.60398E+01**	2.36999E+01	2.36307E+01	2.16508E+01	2.31110E+01
		Std	**6.27315E−01**	1.56816E+00	2.62017E+00	3.45187E+00	3.12823E+00
3	4	Avg	**2.09707E+01**	2.03100E+01	1.92369E+01	1.81455E+01	2.01042E+01
		Std	**6.50360E−01**	8.85478E−01	1.67412E+00	2.12878E+00	1.19604E+00
	5	Avg	**2.20345E+01**	2.11118E+01	2.11664E+01	1.95118E+01	2.14997E+01
		Std	**8.29287E−01**	1.38242E+00	1.07695E+00	3.06477E+00	1.05985E+00
	6	Avg	**2.35609E+01**	2.28875E+01	2.14714E+01	2.06060E+01	2.22108E+01
		Std	1.26560E+00	1.23629E+00	2.52963E+00	2.97349E+00	**1.07281E+00**
4	4	Avg	**1.95144E+01**	1.83250E+01	1.87203E+01	1.82854E+01	1.87655E+01
		Std	**5.78397E−01**	1.16290E+00	1.12936E+00	1.40992E+00	7.41969E−01
	5	Avg	**2.09971E+01**	2.06689E+01	1.96292E+01	1.90553E+01	2.03172E+01
		Std	**9.53233E−01**	1.20685E+00	1.86580E+00	1.92997E+00	1.25738E+00
	6	Avg	**2.18142E+01**	2.10155E+01	2.17088E+01	2.08360E+01	2.15461E+01
		Std	1.18959E+00	1.39560E+00	1.15659E+00	1.45587E+00	**1.11983E+00**
5	4	Avg	**2.11828E+01**	1.99866E+01	1.96157E+01	1.85855E+01	1.99511E+01
		Std	**3.74384E−01**	1.05181E+00	2.32730E+00	3.85444E+00	1.52999E+00
	5	Avg	**2.28259E+01**	2.12287E+01	2.08076E+01	2.01409E+01	2.11083E+01
		Std	**5.84971E−01**	1.58558E+00	3.03975E+00	2.58824E+00	1.55215E+00
	6	Avg	**2.37027E+01**	2.22256E+01	2.29436E+01	2.09776E+01	2.26725E+01
		Std	8.75122E−01	9.55603E−01	9.02340E−01	2.52798E+00	8.53349E−01
6	4	Avg	**2.05187E+01**	1.93676E+01	1.92790E+01	1.97843E+01	2.00219E+01
		Std	**4.40250E−01**	1.42247E+00	1.99952E+00	1.43482E+00	1.41178E+00
	5	Avg	**2.22749E+01**	2.06621E+01	2.11492E+01	1.99631E+01	2.16921E+01
		Std	**7.32708E−01**	1.70130E+00	2.41060E+00	2.19658E+00	1.82422E+00
	6	Avg	**2.39573E+01**	2.20381E+01	2.18827E+01	2.13104E+01	2.26983E+01
		Std	**1.15652E+00**	1.24601E+00	2.85927E+00	3.24219E+00	1.84798E+00
7	4	Avg	**2.14832E+01**	2.03303E+01	2.00058E+01	2.06246E+01	2.07668E+01
		Std	8.20718E−01	1.30639E+00	2.12757E+00	1.26275E+00	1.03948E+00
	5	Avg	**2.31386E+01**	2.14332E+01	2.20011E+01	2.15084E+01	2.18816E+01
		Std	**3.80610E−01**	1.71961E+00	1.52830E+00	1.95182E+00	1.13722E+00
	6	Avg	**2.46541E+01**	2.28779E+01	2.36614E+01	2.24195E+01	2.29500E+01
		Std	**6.82819E−01**	1.35925E+00	1.21547E+00	2.93503E+00	1.22865E+00
8	4	Avg	**2.15768E+01**	2.02150E+01	1.88791E+01	1.72845E+01	2.01260E+01
		Std	**6.05147E−01**	1.53227E+00	2.91852E+00	2.88481E+00	1.76294E+00
	5	Avg	**2.37524E+01**	2.16376E+01	2.17551E+01	1.94681E+01	2.17907E+01
		Std	**7.63915E−01**	1.29217E+00	1.86284E+00	4.42278E+00	1.37690E+00
	6	Avg	**2.51563E+01**	2.29622E+01	2.28298E+01	2.21593E+01	2.37530E+01
		Std	**7.81868E−01**	1.74106E+00	2.65236E+00	2.50298E+00	1.58135E+00

Image	K		SSA	MVO	BLPSO	IWOA
1	4	Avg	1.93736E+01	1.73164E+01	1.65872E+01	1.91582E+01

Table 9 (continued)

Image	K			SSA	MVO	BLPSO	IWOA
			Std	2.38954E+00	2.48317E+00	3.43880E+00	2.23620E+00
	5		Avg	2.04473E+01	1.94653E+01	1.82121E+01	2.06663E+01
			Std	2.90820E+00	2.55223E+00	2.70762E+00	1.82946E+00
	6		Avg	2.12231E+01	1.98379E+01	1.94516E+01	2.03883E+01
			Std	2.45204E+00	2.87713E+00	2.70182E+00	2.99456E+00
2	4		Avg	2.00388E+01	1.90224E+01	1.92072E+01	1.90681E+01
			Std	3.02881E+00	3.08935E+00	1.71742E+00	3.23990E+00
	5		Avg	2.16865E+01	2.06122E+01	1.94363E+01	2.11504E+01
			Std	2.07307E+00	3.27090E+00	3.24533E+00	3.31594E+00
	6		Avg	2.25203E+01	2.15417E+01	1.95151E+01	2.13985E+01
			Std	2.24285E+00	2.30492E+00	2.96904E+00	2.81319E+00
3	4		Avg	1.81294E+01	1.80940E+01	1.88145E+01	1.86217E+01
			Std	2.36710E+00	2.13816E+00	1.56454E+00	2.24503E+00
	5		Avg	2.15307E+01	1.96471E+01	1.81898E+01	2.09464E+01
			Std	1.27278E+00	2.33414E+00	2.03593E+00	1.50150E+00
	6		Avg	2.20457E+01	2.13224E+01	2.03049E+01	2.18566E+01
			Std	1.77083E+00	2.92385E+00	1.87622E+00	1.69399E+00
4	4		Avg	1.88009E+01	1.82409E+01	1.82183E+01	1.79081E+01
			Std	8.62819E-01	1.42564E+00	1.33656E+00	1.49428E+00
	5		Avg	2.03448E+01	1.85133E+01	1.90310E+01	1.87324E+01
			Std	1.06975E+00	2.17185E+00	1.29457E+00	1.94800E+00
	6		Avg	2.06766E+01	2.09358E+01	2.00189E+01	2.06491E+01
			Std	1.42168E+00	1.44592E+00	1.26743E+00	1.46313E+00
5	4		Avg	2.04770E+01	1.94710E+01	1.88619E+01	1.95337E+01
			Std	1.18148E+00	1.46166E+00	1.21587E+00	1.41706E+00
	5		Avg	2.05171E+01	2.04649E+01	1.91728E+01	2.10220E+01
			Std	1.93737E+00	1.36196E+00	1.65815E+00	1.21843E+00
	6		Avg	2.16995E+01	2.02322E+01	2.04868E+01	2.06178E+01
			Std	1.35098E+00	3.18149E+00	1.91024E+00	2.96195E+00
6	4		Avg	1.96300E+01	1.81191E+01	1.88328E+01	1.96261E+01
			Std	2.23192E+00	1.76451E+00	1.82338E+00	1.28504E+00
	5		Avg	2.08428E+01	1.86505E+01	1.95726E+01	2.06325E+01
			Std	2.05018E+00	2.44290E+00	1.62958E+00	1.89450E+00
	6		Avg	2.26555E+01	2.05809E+01	2.06087E+01	2.06027E+01
			Std	2.11833E+00	2.06296E+00	1.95266E+00	2.71420E+00
7	4		Avg	2.07655E+01	1.96927E+01	1.88482E+01	1.92882E+01
			Std	1.09086E+00	1.47783E+00	1.80671E+00	1.49155E+00
	5		Avg	2.16317E+01	2.08343E+01	1.95850E+01	2.07120E+01
			Std	2.33896E+00	1.88821E+00	1.98253E+00	1.86221E+00
	6		Avg	2.26191E+01	2.10131E+01	2.08598E+01	2.23253E+01
			Std	1.27055E+00	2.90536E+00	2.10222E+00	1.92047E+00
8	4		Avg	2.04934E+01	1.95434E+01	1.95148E+01	1.88462E+01
			Std	1.97480E+00	2.28693E+00	1.58634E+00	3.36450E+00
	5		Avg	2.06055E+01	1.88591E+01	1.97820E+01	1.93543E+01
			Std	2.39587E+00	3.28360E+00	1.86045E+00	3.47756E+00
	6		Avg	2.26865E+01	2.11817E+01	2.07919E+01	2.11199E+01
			Std	2.71930E+00	2.75889E+00	2.05991E+00	3.24761E+00

Table 10 The SSIM of each algorithm

Image	K		DMCS	CS	WOA	BA	PSO
1	4	Avg	**7.92140E–01**	6.94955E–01	7.45954E–01	7.57167E–01	7.46926E–01
		Std	**4.94928E–02**	9.93578E–02	8.60484E–02	6.64400E–02	8.67558E–02
	5	Avg	**8.50440E–01**	7.34800E–01	7.81439E–01	7.68807E–01	7.67261E–01
		Std	**2.16982E–02**	7.60575E–02	9.39024E–02	1.22878E–01	9.47175E–02
	6	Avg	**8.74474E–01**	7.91566E–01	8.11687E–01	8.09940E–01	7.96511E–01
		Std	**2.05488E–02**	6.03800E–02	6.95099E–02	9.74572E–02	1.00818E–01
2	4	Avg	**8.18274E–01**	7.81004E–01	7.47595E–01	7.36751E–01	7.69098E–01
		Std	**2.25541E–02**	4.82128E–02	1.29120E–01	7.59018E–02	1.19653E–01
	5	Avg	**8.74334E–01**	7.91573E–01	8.00231E–01	7.74973E–01	8.41665E–01
		Std	**1.80578E–02**	6.03766E–02	8.29076E–02	7.33877E–02	3.60211E–02
	6	Avg	**9.09815E–01**	8.44943E–01	8.51560E–01	8.02178E–01	8.38138E–01
		Std	**1.47404E–02**	4.39596E–02	7.01491E–02	9.31510E–02	8.50565E–02
3	4	Avg	**7.47795E–01**	7.40978E–01	7.03870E–01	6.64628E–01	7.39145E–01
		Std	**3.61903E–02**	3.76937E–02	6.05180E–02	9.68681E–02	4.41265E–02
	5	Avg	**7.88009E–01**	7.54594E–01	7.73819E–01	7.18394E–01	7.77812E–01
		Std	**3.79279E–02**	5.11270E–02	2.95956E–02	1.07341E–01	3.82315E–02
	6	Avg	**8.32435E–01**	8.08926E–01	7.80232E–01	7.61146E–01	7.98609E–01
		Std	4.53010E–02	**3.63892E–02**	6.55516E–02	8.85763E–02	2.20132E–02
4	4	Avg	**7.77085E–01**	7.41303E–01	7.55885E–01	7.38146E–01	7.56758E–01
		Std	**2.42157E–02**	4.97199E–02	4.24257E–02	5.75097E–02	2.61947E–02
	5	Avg	**8.19605E–01**	8.02788E–01	7.81411E–01	7.63365E–01	8.01182E–01
		Std	3.16446E–02	3.16690E–02	5.83017E–02	5.92865E–02	3.70661E–02
	6	Avg	**8.44434E–01**	8.11004E–01	8.32491E–01	8.12420E–01	8.33249E–01
		Std	**2.07765E–02**	3.69416E–02	3.26889E–02	4.55376E–02	2.52839E–02
5	4	Avg	**8.05991E–01**	7.60933E–01	7.55085E–01	7.11598E–01	7.63404E–01
		Std	**1.12089E–02**	3.39289E–02	8.19980E–02	1.47460E–01	5.17084E–02
	5	Avg	**8.51374E–01**	7.89373E–01	7.85386E–01	7.59657E–01	7.98270E–01
		Std	**1.55282E–02**	6.30391E–02	1.27387E–01	1.16309E–01	5.14605E–02
	6	Avg	**8.73301E–01**	8.24291E–01	8.51752E–01	7.92219E–01	8.46070E–01
		Std	**2.02222E–02**	3.19812E–02	2.26476E–02	9.92581E–02	2.21750E–02
6	4	Avg	**8.55088E–01**	8.20272E–01	8.19133E–01	8.33280E–01	8.38091E–01
		Std	**1.24535E–02**	4.34575E–02	5.82404E–02	3.77894E–02	4.05866E–02
	5	Avg	**8.92839E–01**	8.32433E–01	8.54395E–01	8.29597E–01	8.72426E–01
		Std	**1.87752E–02**	5.02405E–02	5.52819E–02	6.49229E–02	4.76097E–02
	6	Avg	**9.09004E–01**	8.51596E–01	8.54425E–01	8.42056E–01	8.88500E–01
		Std	**2.08462E–02**	4.40681E–02	1.29661E–01	1.30040E–01	3.60453E–02
7	4	Avg	**8.07837E–01**	7.67315E–01	7.69668E–01	7.87461E–01	7.88628E–01
		Std	**2.75355E–02**	4.10239E–02	6.19526E–02	3.47564E–02	2.95690E–02
	5	Avg	**8.59768E–01**	7.86385E–01	8.22571E–01	8.08752E–01	8.16683E–01
		Std	**8.80190E–03**	5.19892E–02	3.38030E–02	5.55757E–02	3.21082E–02
	6	Avg	**8.89313E–01**	8.19579E–01	8.60850E–01	8.21441E–01	8.44256E–01
		Std	**1.17102E–02**	5.18458E–02	2.67373E–02	9.02119E–02	3.21252E–02
8	4	Avg	**8.29909E–01**	7.74505E–01	7.48257E–01	6.89417E–01	7.86626E–01
		Std	**2.05435E–02**	6.20769E–02	8.88404E–02	1.14192E–01	5.04410E–02
	5	Avg	**8.80583E–01**	8.03986E–01	8.31068E–01	7.46942E–01	8.32353E–01
		Std	**2.00778E–02**	5.27814E–02	4.57506E–02	1.90413E–01	3.76893E–02
	6	Avg	**9.05047E–01**	8.41290E–01	8.50318E–01	8.38592E–01	8.72948E–01
		Std	**1.56370E–02**	6.34576E–02	7.67947E–02	6.24977E–02	4.19552E–02

Image	K		SSA	MVO	BLPSO	IWOA
1	4	Avg	7.42741E–01	6.82694E–01	6.49225E–01	7.35018E–01

Table 10 (continued)

Image	K			SSA	MVO	BLPSO	IWOA
		Std		7.28488E–02	7.59807E–02	1.15304E–01	6.52832E–02
	5	Avg		7.63603E–01	7.31174E–01	6.87785E–01	7.74343E–01
		Std		8.72299E–02	8.48116E–02	8.29148E–02	5.90950E–02
	6	Avg		7.87056E–01	7.45616E–01	7.30285E–01	7.62871E–01
		Std		7.03595E–02	9.19439E–02	7.32925E–02	9.26546E–02
2	4	Avg		7.55348E–01	7.02824E–01	7.06520E–01	7.25087E–01
		Std		9.93601E–02	1.07322E–01	7.26523E–02	9.81276E–02
	5	Avg		8.03133E–01	7.62580E–01	7.09765E–01	7.81465E–01
		Std		5.58742E–02	9.68111E–02	1.26908E–01	9.94545E–02
	6	Avg		8.16302E–01	7.65473E–01	7.12001E–01	7.84280E–01
		Std		6.00467E–02	9.87799E–02	1.00161E–01	8.23673E–02
3	4	Avg		6.74217E–01	6.62789E–01	6.89728E–01	6.94301E–01
		Std		7.92452E–02	7.29725E–02	6.73252E–02	8.67020E–02
	5	Avg		7.78748E–01	7.20334E–01	6.74354E–01	7.68370E–01
		Std		3.20260E–02	7.14880E–02	6.65393E–02	4.87774E–02
	6	Avg		7.87983E–01	7.59276E–01	7.26739E–01	7.93206E–01
		Std		5.14165E–02	9.35993E–02	6.32935E–02	4.63834E–02
4	4	Avg		7.53391E–01	7.32783E–01	7.33485E–01	7.24056E–01
		Std		3.17617E–02	4.50181E–02	5.31553E–02	5.65658E–02
	5	Avg		7.97821E–01	7.31476E–01	7.47722E–01	7.53765E–01
		Std		**2.87091E–02**	6.77956E–02	4.22274E–02	6.30489E–02
	6	Avg		7.95808E–01	7.99182E–01	7.70218E–01	8.04617E–01
		Std		4.84239E–02	4.26008E–02	4.90007E–02	4.05159E–02
5	4	Avg		7.80838E–01	7.33454E–01	7.18863E–01	7.46587E–01
		Std		4.09052E–02	6.42353E–02	4.99082E–02	4.96605E–02
	5	Avg		7.77866E–01	7.63274E–01	7.13216E–01	7.95566E–01
		Std		7.10073E–02	6.06028E–02	6.71731E–02	4.46452E–02
	6	Avg		8.13664E–01	7.56151E–01	7.66299E–01	7.76245E–01
		Std		4.10760E–02	1.05444E–01	6.20045E–02	9.66625E–02
6	4	Avg		8.24686E–01	7.69328E–01	7.94437E–01	8.24036E–01
		Std		6.77075E–02	5.92267E–02	5.85605E–02	3.63884E–02
	5	Avg		8.48549E–01	7.67146E–01	8.11797E–01	8.38976E–01
		Std		5.80580E–02	9.06532E–02	4.84429E–02	5.03692E–02
	6	Avg		8.75727E–01	8.27553E–01	8.30827E–01	8.28170E–01
		Std		5.39845E–02	6.19966E–02	5.03269E–02	9.96500E–02
7	4	Avg		7.87814E–01	7.46673E–01	7.14272E–01	7.41741E–01
		Std		3.48411E–02	5.15399E–02	5.96316E–02	5.15038E–02
	5	Avg		8.04715E–01	7.76373E–01	7.23577E–01	7.79440E–01
		Std		7.35891E–02	5.71046E–02	7.34253E–02	5.42636E–02
	6	Avg		8.25744E–01	7.70038E–01	7.62930E–01	8.14914E–01
		Std		3.97364E–02	1.02141E–01	6.98593E–02	5.88067E–02
8	4	Avg		7.87309E–01	7.39000E–01	7.46774E–01	7.39405E–01
		Std		8.53396E–02	1.32680E–01	5.79216E–02	1.07437E–01
	5	Avg		7.94291E–01	7.03662E–01	7.44580E–01	7.55012E–01
		Std		6.93470E–02	1.27444E–01	7.34107E–02	1.11183E–01
	6	Avg		8.42523E–01	7.99564E–01	7.79749E–01	7.93966E–01
		Std		8.17797E–02	7.95253E–02	7.21412E–02	9.36475E–02

Table 11 Wilcoxon signed-rank test result for each algorithm on FSIM

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	+ / - / =	~	4/0/4	6/0/2	5/0/3	2/0/6	3/0/5	8/0/8	8/0/0	7/0/1
	Avg	**1**	4.625	5.625	5.625	2.75	3.75	7.375	7.5	6.75
	Rank	**1**	4	5	5	2	3	8	9	7
5	+ / - / =	~	4/0/4	6/0/2	6/0/2	3/0/5	4/0/4	8/0/0	8/0/0	7/0/1
	Avg	**1.25**	4.5	4.375	7.375	2.875	3.625	7.625	8.25	5.125
	Rank	**1**	5	4	7	2	3	8	9	6
6	+ / - / =	~	5/0/3	4/0/4	7/0/1	5/0/3	5/0/3	7/0/1	8/0/0	7/0/1
	Avg	**1.5**	4.25	3.375	5.875	2.625	4.375	7.75	8.125	7.125
	Rank	**1**	4	3	6	2	5	8	9	7

Table 12 Wilcoxon signed-rank test result for each algorithm on PSNR

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	+ / - / =	~	8/0/0	7/0/1	6/0/2	5/0/3	7/0/1	8/0/0	8/0/0	8/0/0
	Avg	**1**	4.125	5.25	5.75	3	3.75	7.625	7.5	7
	Rank	**1**	4	5	6	2	3	9	8	7
5	+ / - / =	~	7/0/1	7/0/1	8/0/0	7/0/1	7/0/1	8/0/0	8/0/0	8/0/0
	Avg	**1**	4.25	3.625	6.5	3.125	4.125	8	8.25	6.125
	Rank	**1**	5	3	7	2	4	8	9	6
6	+ / - / =	~	7/0/1	7/0/1	8/0/0	7/0/1	7/0/1	8/0/0	8/0/0	8/0/0
	Avg	**1**	3.5	3.375	5.75	3	5	7.5	8.625	7.25
	Rank	**1**	4	3	6	2	5	8	9	7

Table 13 Wilcoxon signed-rank test result for each algorithm on SSIM

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	+ / - / =	~	7/0/1	6/0/2	6/0/2	4/0/4	7/0/1	8/0/0	8/0/0	8/0/0
	Avg	**1**	4.5	4.75	5.875	2.625	3.875	8.125	7.625	6.625
	Rank	**1**	4	5	6	2	3	9	8	7
5	+ / - / =	~	8/0/0	7/0/1	8/0/0	7/0/1	7/0/1	8/0/0	8/0/0	7/0/1
	Avg	**1**	5	3.5	6.375	2.75	4.375	8.125	8.625	5.25
	Rank	**1**	5	3	7	2	4	8	9	6
6	+ / - / =	~	8/0/0	7/0/1	8/0/0	7/0/1	8/0/0	8/0/0	8/0/0	8/0/0
	Avg	**1**	4.375	3	5.375	2.875	5	8	8.625	6.75
	Rank	**1**	4	3	6	2	5	8	9	7

Table 14 Friedman test result for each algorithm on FSIM

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	Avg	**6.8**	5.1375	4.9125	4.6625	5.66875	5.4875	3.88125	3.99375	4.45625
	Rank	**1**	4	5	6	2	3	9	8	7
5	Avg	7.0125	5.325	5.24375	4.5375	5.8375	5.63125	3.675	3.1625	4.575
	Rank	1	4	5	7	2	3	8	9	6
6	Avg	6.99375	5.10625	5.56875	4.75	5.8625	5.24375	3.925	3.225	4.325
	Rank	1	5	3	6	2	4	8	9	7

Table 15 Friedman test result for each algorithm on PSNR

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	Avg	7.4625	5.18125	4.9625	4.4375	5.6375	5.375	3.93125	3.7	4.3125
	Rank	1	4	5	6	2	3	8	9	7
5	Avg	7.8375	5.16875	5.4125	4.35625	5.81875	5.26875	3.625	3.05	4.4625
	Rank	1	5	3	7	2	4	8	9	6
6	Avg	7.95	5.16875	5.8	4.7	5.66875	4.85	3.81875	2.8625	4.18125
	Rank	1	4	2	6	3	5	8	9	7

Table 16 Friedman test result for each algorithm on SSIM

K		DMCS	CS	WOA	BA	PSO	SSA	MVO	BLPSO	IWOA
4	Avg	7.28125	5.09375	5.19375	4.64375	5.6625	5.46875	3.78125	3.51875	4.35625
	Rank	1	5	4	6	2	3	8	9	7
5	Avg	7.9625	4.73125	5.5875	4.70625	5.975	5.21875	3.45625	2.825	4.5375
	Rank	1	5	3	6	2	4	8	9	7
6	Avg	8.075	4.8375	6.025	4.9375	5.96875	4.8	3.46875	2.7	4.1875
	Rank	1	5	2	4	3	6	8	9	7

$$Rt_\alpha^n = \frac{1}{1-\alpha} ln \sum_{i=t_{n-1}+1}^{t_n} \left(\frac{p(i)}{w_n(t)} \right)^\alpha, n = 1 \dots N. \quad (16)$$

The selected optimal threshold by the algorithm should meet Eq. (17):

$$[t_1, t_2, \dots, t_N] = arg \max \left(\sum_{n=1}^N Rt_\alpha^n(t_n) \right). \quad (17)$$

4.3 Evaluation Metrics for Image Segmentation

This subsection details three commonly used image quality evaluation metrics: FSIM, PSNR, and SSIM. They evaluate the image feature similarity, image distortion degree, and image structural integrity.

(1) FSIM is an image quality metric based on a low-level feature, which uses both Phase Congruency (PC) and the Gradient Magnitude (GM) [85] features. PC and GM indicate complementary aspects of the visual quality of an image. Below we will explain Eqs. (18)–(21).

PC_1 and PC_2 represent the PC map extracted from images f_1 and f_2, respectively, and G_1 and G_2 represent the GM maps extracted from them. The similarity measure $S_{PC}(x)$ for $PC_1(x)$ and $PC_2(x)$ is calculated by Eq. (18), and the similarity measure $S_G(x)$ is defined as Eq. (19):

$$S_{PC}(x) = \frac{2PC_1(x) \cdot PC_2(x) + T_1}{PC_1^2(x) + PC_2^2(x) + T_1}, \quad (18)$$

$$S_G(x) = \frac{2G_1(x) \cdot G_2(x) + T_2}{G_1^2(x) + G_2^2(x) + T_2}, \quad (19)$$

where T_1, T_2 both are positive constants, and T_1 plays a role in enhancing the stability of S_{PC}. The dynamic range of GM values determines the value of T_2. Define S_L as the similarity between f_1 and f_2, which is calculated by combining S_{PC} and S_G, and the calculation method is shown in Eq. (20):

$$S_L(x) = [S_{PC}(x)]^\alpha \cdot [S_G(x)]^\beta, \quad (20)$$

where α and β are relative importance parameters that can be used to adjust the PC and GM values. From the above description. The FSIM index between f_1 and f_2 can be defined as Eq. (21).

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x) \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)}, \quad (21)$$

where Ω denotes the whole image space domain, and PC_m is obtained through equation $\max(PC_1(x), PC_2(x))$.

(2) PSNR is a simple and commonly used fidelity measurement method and has traditionally been used in analog systems as a consistent quality metric. Due to its mathematical convenience and simplicity, it is an attractive measure of image quality (loss) [84, 98].

The mathematical formula for PSNR is as follows:

$$PSNR(x, y) = 10 \log_{10} \frac{255^2}{MSE(x, y)}, \quad (22)$$
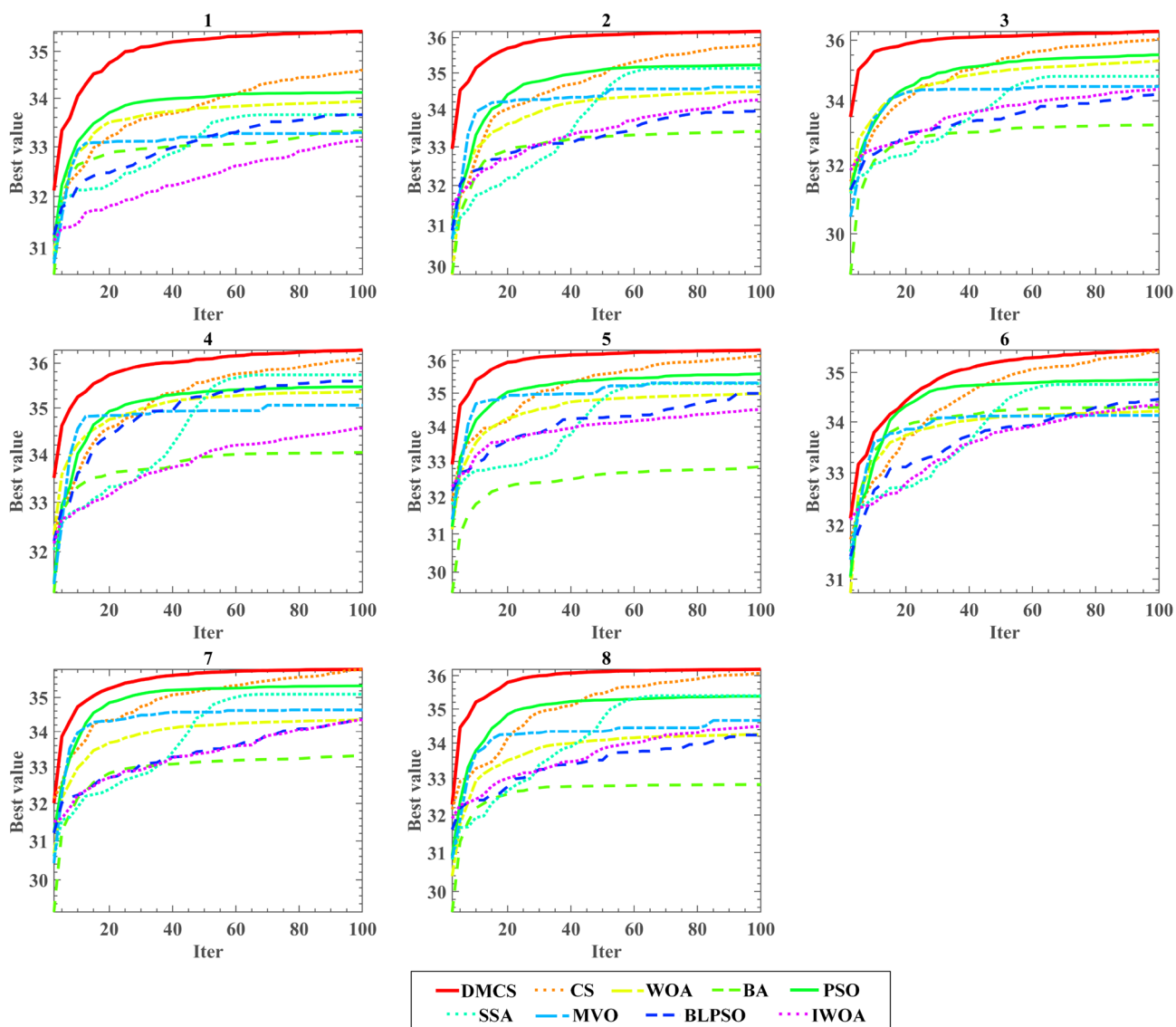
Fig. 11 2D Rényi entropy convergence curves for eight images at the threshold of 4

$$\text{MES}(x, y) = \frac{1}{N} \sum_{I=1}^{N} (x_i - y_i)^2, \qquad (23)$$

where x and y denote the vector of reference and test image signals, respectively. $\text{MES}(x, y)$ means the mean-squared error between x and y. N denotes the total number of pixels in the image [99].

(3) SSIM is a metric to measure the quality by the similarity of the captured images. The SSIM index is obtained by three aspects of similarity: luminance, contrast, and structure. Defined x and y are two images, the luminance comparison function $l(x, y)$ is defined as Eq. (24):

$$l(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \qquad (24)$$

where μ means the mean value of the image, and C_1 is a constant to avoid instability when $\mu_x^2 + \mu_y^2$ is very close to zero. Similarity, the contrast comparison function $c(x, y)$ can be defined as Eq. (25):

$$c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \qquad (25)$$
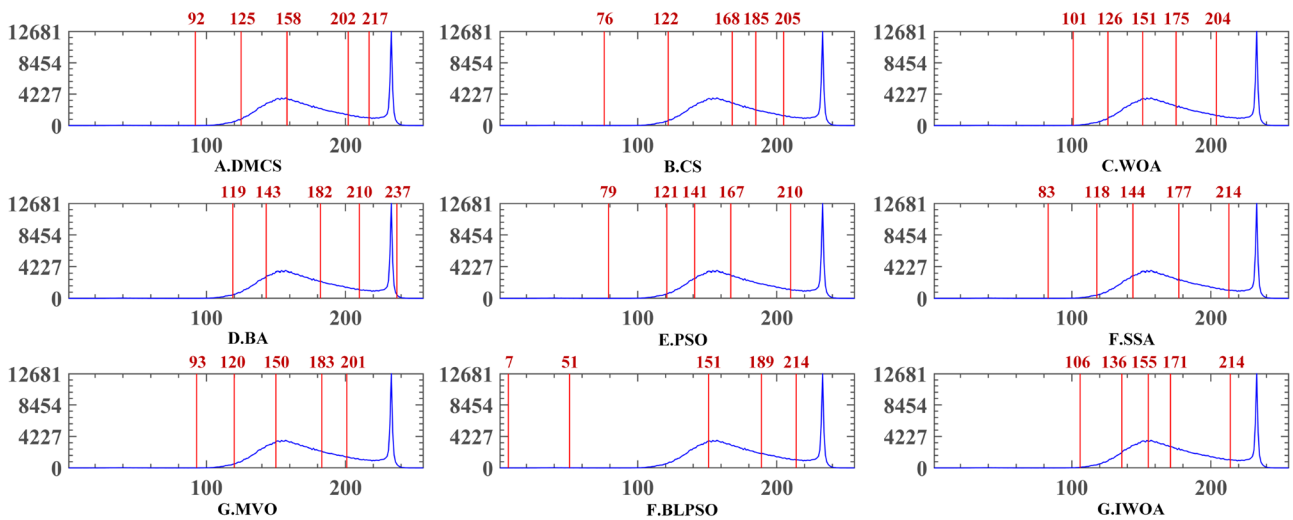
Fig. 12 Thresholds for image 1 obtained by each algorithm at level 5

where σ is the standard deviation of two contrasting images, and C_2 is a constant. The structure comparison function $s(x, y)$ is calculated by Eq. (26):

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}. \tag{26}$$

The mathematical formula for SSIM is as follows:

$$\text{SSIM}(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{27}$$

where set $C_3 = \frac{1}{2}C_2$ according to [86].

4.4 Experimental Result Analyses

The original color images and 2D histograms of the eight images used in the experiments are shown in Fig. 10. Column A in Fig. 10 is the color image of the original images, and column B represents the 2D histogram generated by combining the NLM graph and grayscale image. Tables 8, 9 and 10 show the results of the analysis of the experimental results using the image quality evaluation metrics, where "K" is the threshold value, "Avg" denotes the average, and "Std" means the standard deviation. According to the results of FSIM, the DMCS algorithm achieves the best results on all eight images at a threshold of 4; at thresholds of 5 and 6, it fails to achieve the optimal Avg or Std only on images 2, 4 and 4, 5. This indicates that most of the images segmented using the DMCS algorithm are closer to the test images than the competitors. According to the results of PSNR, the DMCS algorithm performs well on all eight images at thresholds 4, and 5, and achieves the maximum Avg and the minimum Std on images except images 3 and 4 at threshold 6. This indicates that most of the images segmented using the DMCS algorithm have less distortion than the competitors. According to the results of SSIM, the DMCS algorithm achieves the best results on all eight images at a threshold of 4. At a threshold of 5 and 6, it fails to achieve the optimal Std only on images 4 and 3. This indicates that most of the images segmented using the DMCS algorithm are more structurally complete than the competitors. Therefore, DMCS can effectively avoid stagnation of the algorithm and is more stable. In addition, the accuracy of image segmentation improves as the threshold increases. The Wilcoxon signed-rank test was performed on the results of Tables 8, 9 and 10 to obtain Tables 11 , 12 and 13, and the best results are bolded. Based on the results of " + / - / = ", it can be found that DMCS significantly outperforms the comparison algorithm on 8 images. Therefore, DMCS ranks first overall and has the best performance. Tables 14, 15 and 16 show the results of further analysis with the Friedman test, where the DMCS algorithm gets the largest average and the first ranking with a large advantage on all thresholds. In addition, Fig. 11 shows the convergence curves of

DMCS:Segmented gray and Jet colormap images

CS:Segmented gray amd Jet colormap images

WOA:Segmented gray amd Jet colormap images

BA:Segmented gray and Jet colormap images

PSO:Segmented gray and Jet colormap images

SSA:Segmented gray and Jet colormap images

MVO:Segmented gray and Jet colormap images

BLPSO:Segmented gray and Jet colormap images

IWOA:Segmented gray and Jet colormap images

Fig. 13 Segmentation results for each method at threshold level 6 of image 4

the DMCS algorithm and the contrast algorithm for the two-dimensional Rényi entropy for a threshold of 4 in 8 images; the larger the value of the curve convergence, the more information it contains about the image. The figure shows that the DMCS algorithm has the fastest convergence speed and accuracy. Figure 12 offers the best thresholds of the DMCS algorithm and its contrast algorithm for a threshold of 5 on image 1. The best thresholds have been marked in red font on the image, and the comparison reveals that the DMCS algorithm can over-obtain reasonable thresholds. Figure 13 visualizes the images after the segmentation of image 4 at a threshold of 6 for each algorithm in the experiment, the first image in each group is the segmented grayscale image, and the second is the colormap map. It can be seen from the segmented grayscale images that the DMCS algorithm has more transparent and more complete details in the segmented images compared with the CS algorithm, and the results are not inferior compared with other comparison algorithms.

5 Conclusions and Future Works

This paper proposes a multi-threshold image segmentation model combining NLM 2D histogram and an improved CS algorithm to segment LN images. To improve the defects of the original CS algorithm and increase the efficiency of image segmentation, an enhanced algorithm named DMCS is proposed by introducing the DM and AβHC strategies in the CS algorithm. Introducing the DM and AβHC strategies increases the cuckoo population diversity and helps search for better solutions. DMCS is tested on the CEC 2017 dataset. Among the mechanism comparison, experimental results prove that the DM strategy can accelerate the algorithm's convergence. In contrast, the AβHC strategy is beneficial to improve the quality and accuracy of the solution, and the experimental results with the traditional and advanced algorithms also show that the proposed improved algorithm performs better function optimization. Finally, the ability of DMCS to solve practical problems is verified on real LN image segmentation. However, the proposed method has some limitations. Adding the mechanisms inevitably increases the algorithm's time complexity, which can be improved by parallel computing and high-performance computing techniques in the future. In addition, for other optimization problems, such as feature selection and energy parameter optimization with different search space characteristics and objectives, the optimization effect of the DMCS algorithm will be affected. The DMCS algorithm can continue to be improved in the next step to be applied to more cases, such as optimizing machine learning models.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest regarding the publication of the article.

References

1. Huo, Y. K., Deng, R. N., Liu, Q., Fogo, A. B., & Yang, H. C. (2021). AI applications in renal pathology. *Kidney International, 99*, 1309–1320.

2. Danila, M. I., Pons-Estel, G. J., Zhang, J., Vila, L. M., Reveille, J. D., & Alarcon, G. S. (2009). Renal damage is the most important predictor of mortality within the damage index: Data from LUMINA LXIV, a multiethnic US cohort. *Rheumatology (Oxford), 48*, 542–545.

3. James, J. A., Guthridge, J. M., Chen, H., Lu, R., Bourn, R. L., Bean, K., Munroe, M. E., Smith, M., Chakravarty, E., Baer, A. N., Noaiseh, G., Parke, A., Boyle, K., Keyes-Elstein, L., Coca, A., Utset, T., Genovese, M. C., Pascual, V., Utz, P. J., … St Clair, E. W. (2020). Unique Sjögren's syndrome patient subsets defined by molecular features. *Rheumatology (Oxford), 59*, 860–868.

4. Fanouriakis, A., Kostopoulou, M., Cheema, K., Anders, H. J., Aringer, M., Bajema, I., Boletis, J., Frangou, E., Houssiau, F. A., Hollis, J., Karras, A., Marchiori, F., Marks, S. D., Moroni, G., Mosca, M., Parodis, I., Praga, M., Schneider, M., Smolen, J. S., … Boumpas, D. T. (2020). 2019 Update of the Joint European League Against Rheumatism and European Renal Association-European Dialysis and Transplant Association (EULAR/ERA-EDTA) recommendations for the management of lupus nephritis. *Annals of the Rheumatic Diseases, 79*, 713–723.

5. Liu, Z. J., Su, W., Ao, J. P., Wang, M., Jiang, Q. L., He, J., Gao, H., Lei, S., Nie, J. S., Yan, X. F., Guo, X. J., Zhou, P. H., Hu, H., & Ji, M. B. (2022). Instant diagnosis of gastroscopic biopsy via deep-learned single-shot femtosecond stimulated Raman histology. *Nature Communications, 13*, 4050.

6. Jin, K., Huang, X. R., Zhou, J. X., Li, Y. X., Yan, Y., Sun, Y. B., Zhang, Q. N., Wang, Y. Q., & Ye, J. (2022). FIVES: A fundus image dataset for artificial intelligence based vessel segmentation. *Scientific Data, 9*, 475.

7. Li, Q. H., Song, D. Q., Yuan, C. M., & Nie, W. (2022). An image recognition method for the deformation area of open-pit rock slopes under variable rainfall. *Measurement, 188*, 110544.

8. Kline, A., Chung, H. J., Rahmani, W., & Chun, J. (2021). Semi-supervised segmentation of renal pathology: an alternative to manual segmentation and input to deep learning training. In: *IEEE Engineering in Medicine and Biology Society. Annual International Conference,* Mexico, pp. 2688–2691.

9. Albayrak, A., & Bilgin, G. (2019). Automatic cell segmentation in histopathological images via two-staged superpixel-based algorithms. *Medical & Biological Engineering & Computing, 57*, 653–665.

10. Yoruk, U., Hargreaves, B. A., & Vasanawala, S. S. (2018). Automatic renal segmentation for MR urography using 3D-GrabCut and random forests. *Magnetic Resonance in Medicine, 79*, 1696–1707.

11. Gadermayr, M., Eschweiler, D., Jeevanesan, A., Klinkhammer, B. M., Boor, P., & Merhof, D. (2017). Segmenting renal whole slide images virtually without training data. *Computers in Biology and Medicine, 90*, 88–97.

12. Li, X. L., Chen, X. J., Yao, J. H., Zhang, X., & Tian, J. (2011). Renal cortex segmentation using optimal surface search with novel graph construction. *Medical Image Computing Computing Assisted Intervention, 14*, 387–394.

13. Yang, X., Le Minh, H., Tim Cheng, K. T., Sung, K. H., & Liu, W. (2016). Renal compartment segmentation in DCE-MRI images. *Medical Image Analysis, 32*, 269–280.

14. Luo, J., Yang, Y. S., & Shi, B. Y. (2019). Multi-threshold Image segmentation of 2D Otsu based on improved adaptive differential evolution algorithm. *Journal of Electronics & Information Technology, 41*, 2017–2024.

15. Nguyen, T. T., Wang, H. J., Dao, T. K., Pan, J. S., Ngo, T. G., & Yu, J. (2020). A scheme of color image multithreshold segmentation based on improved moth-flame algorithm. *Ieee Access, 8*, 174142–174159.

16. Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249.

17. Zhou, Y. Q., Yang, X., Ling, Y., & Zhang, J. Z. (2018). Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation. *Multimedia Tools and Applications, 77*, 23699–23727.

18. Zhao, D., Liu, L., Yu, F. H., Heidari, A. A., Wang, M. J., Oliva, D., Muhammad, K., & Chen, H. L. (2021). Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Systems with Applications, 167*, 114122.

19. Sharma, S., Saha, A. K., Majumder, A., & Nama, S. (2021). MPBOA - A novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimedia Tools and Applications, 80*, 12035–12076.

20. Chakraborty, S., Sharma, S., Saha, A. K., & Saha, A. (2022). A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artificial Intelligence Review, 55*, 4605–4716.

21. Chakraborty, S., Saha, A. K., Nama, S., & Debnath, S. (2021). COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Computers in Biology and Medicine, 139*, 104984.

22. Li, R. H., Wu, X. L., Tian, H., Yu, N., & Wang, C. (2022). Hybrid Memetic pretrained factor analysis-based deep belief networks for transient electromagnetic inversion. *Ieee Transactions on Geoscience and Remote Sensing, 60*, 1–20.

23. Duan, H. B., & Luo, Q. N. (2015). New progresses in swarm intelligence-based computation. *International Journal of Bio-Inspired Computation, 7*, 26–35.

24. Lin, Y., Song, H., Ke, F., Yan, W. Z., Liu, Z. K., & Cai, F. M. (2022). Optimal caching scheme in D2D networks with multiple robot helpers. *Computer Communications, 181*, 132–142.

25. Chen, W. N., Zhang, J., Lin, Y., Chen, N., Zhan, Z. H., Chung, H. S. H., Li, Y., & Shi, Y. H. (2013). Particle swarm optimization with an aging leader and challengers. *Ieee Transactions on Evolutionary Computation, 17*, 241–258.

26. Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

27. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69*, 46–61.

28. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*, 28–39.

29. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems-the International Journal of Escience, 97*, 849–872.

30. Mirjalili, S., & Lewis, A. (2016). The Whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

31. Holland, J. H. (1992). Genetic algorithms. *Scientific American, 267*, 66–72.

32. Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*, 17–35.

33. Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.

34. Gharehchopogh, F. S. (2022). An improved tunicate swarm algorithm with best-random mutation strategy for global optimization problems. *Journal of Bionic Engineering, 19*, 1177–1202.

35. Gharehchopogh, F. S. (2022). Advances in tree seed algorithm: A comprehensive survey. *Archives of Computational Methods in Engineering, 29*, 3281–3304.

36. Kiran, M. S. (2015). TSA: Tree-seed algorithm for continuous optimization. *Expert Systems with Applications, 42*, 6686–6698.

37. Xue, J. K., & Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. *Systems Science & Control Engineering, 8*, 22–34.

38. Gharehchopogh, F. S., Namazi, M., Ebrahimi, L., & Abdollahzadeh, B. (2022). Advances in Sparrow Search Algorithm: A comprehensive survey. *Archives of Computational Methods in Engineering, 30*, 427–455.

39. Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures, 139*, 98–112.

40. Gharehchopogh, F. S., Shayanfar, H., & Gholizadeh, H. (2020). A comprehensive survey on symbiotic organisms search algorithms. *Artificial Intelligence Review, 53*, 2265–2312.

41. Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation, 48*, 1–24.

42. Sahoo, S. K., & Saha, A. K. (2022). A hybrid moth flame optimization algorithm for global optimization. *Journal of Bionic Engineering, 19*, 1522–1543.

43. Awadallah, M. A., Hammouri, A. I., Al-Betar, M. A., Braik, M. S., & Elaziz, M. A. (2022). Binary Horse herd optimization algorithm with crossover operators for feature selection. *Computers in Biology and Medicine, 141*, 105152.

44. Dong, R. Y., Chen, H. L., Heidari, A. A., Turabieh, H., Mafarja, M., & Wang, S. S. (2021). Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowledge-Based Systems, 233*, 107529.

45. Liu, Y., Heidari, A. A., Cai, Z. N., Liang, G. X., Chen, H. L., Pan, Z. F., Alsufyani, A., & Bourouis, S. (2022). Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. *Neurocomputing, 503*, 325–362.

46. Piri, J., & Mohapatra, P. (2021). An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection. *Computers in Biology and Medicine, 135*, 104558.

47. Zhang, Y. N., Liu, R. J., Heidari, A. A., Wang, X., Chen, Y., Wang, M. J., & Chen, H. L. (2021). Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing, 430*, 185–212.

48. Yang, X. S., & Deb, S. (2014). Cuckoo search: Recent advances and applications. *Neural Computing & Applications, 24*, 169–174.

49. Fan, J. Y., Xu, W. J., Huang, Y., & Samuel, R. D. J. (2021). Application of Chaos Cuckoo Search algorithm in computer vision technology. *Soft Computing, 25*, 12373–12387.

50. Long, W., Zhang, W. Z., Huang, Y. F., & Chen, Y. X. (2014). A hybrid cuckoo search algorithm with feasibility-based rule for constrained structural optimization. *Journal of Central South University, 21*, 3197–3204.

51. Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied Soft Computing, 19*, 93–101.

52. Rosli, R., & Mohamed, Z. (2021). Optimization of modified Bouc-Wen model for magnetorheological damper using modified cuckoo search algorithm. *Journal of Vibration and Control, 27*, 1956–1967.

53. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Ieee Transactions on Evolutionary Computation, 1*, 67–82.

54. Wu, G. H., Mallipeddi, R., & Suganthan, P. (2016). *Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization*, Nanyang Technological University, Singapore, Tech. Rep, 1–18.

55. Fister, I., Yang, X. S., Fister, D., & Fister, I. (2014). Cuckoo search: A brief literature review. In X.-S. Yang (Ed.), *Cuckoo search and firefly algorithm: Theory and applications* (pp. 49–62). Springer International Publishing.

56. Al-Betar, M. A., Aljarah, I., Awadallah, M. A., Faris, H., & Mirjalili, S. (2019). Adaptive β-hill climbing for optimization. *Soft Computing, 23*, 13489–13512.

57. Al-Betar, M. A. (2017). β-Hill climbing: An exploratory local search. *Neural Computing & Applications, 28*, 153–168.

58. Li, R. H., Yu, N., Wang, X. B., Liu, Y., Cai, Z. K., & Wang, E. C. (2022). Model-based synthetic geoelectric sampling for magnetotelluric inversion with deep neural networks. *Ieee Transactions on Geoscience and Remote Sensing, 60*, 1–14.

59. Liao, L. Y., Du, L., & Guo, Y. C. (2022). Semi-supervised SAR target detection based on an improved faster R-CNN. *Remote Sensing, 14*, 143.

60. Liu, R. J., Wang, X. S., Lu, H. M., Wu, Z. H., Fan, Q., Li, S. X., & Jin, X. (2021). SCCGAN: Style and characters inpainting based on CGAN. *Mobile Networks & Applications, 26*, 3–12.

61. Yang, B., Xu, S. Y., Chen, H. R., Zheng, W. F., & Liu, C. (2022). Reconstruct dynamic soft-tissue with stereo endoscope based on a single-layer network. *Ieee Transactions on Image Processing, 31*, 5828–5840.

62. Wang, K. N., Zhang, B. L., Alenezi, F., & Li, S. M. (2022). Communication-efficient surrogate quantile regression for non-randomly distributed system. *Information Sciences, 588*, 425–441.

63. Dong, C. H., Li, Y. H., Gong, H. F., Chen, M. X., Li, J. X., Shen, Y., & Yang, M. (2022). A survey of natural language generation. *ACM Computing Surveys, 55*, 173.

64. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*, 675–701.

65. Li, S. M., Chen, H. L., Wang, M. J., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems-the International Journal of Escience, 111*, 300–323.

66. Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Computing & Applications, 27*, 495–513.

67. Wang, Y. J., & Chen, Y. (2020). An improved farmland fertility algorithm for global function optimization. *Ieee Access, 8*, 111850–111874.

68. Heidari, A. A., Abbaspour, R. A., & Chen, H. L. (2019). Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. *Applied Soft Computing, 81*, 105521.

69. Qu, C. W., Zeng, Z. L., Dai, J., Yi, Z. J., & He, W. (2018). A modified sine-cosine algorithm based on neighborhood search and greedy Levy mutation. *Computational Intelligence and Neuroscience, 2018*, 4231647.

70. Zhu, A. J., Xu, C. P., Li, Z., Wu, J., & Liu, Z. B. (2015). Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. *Journal of Systems Engineering and Electronics, 26*, 317–328.

71. Chen, H., Heidari, A. A., Zhao, X. H., Zhang, L. J., & Chen, H. L. (2020). Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies. *Expert Systems with Applications, 144*, 113113.

72. Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *Ieee Transactions on Evolutionary Computation, 10*, 281–295.

73. Ling, Y., Zhou, Y. Q., & Luo, Q. F. (2017). Levy flight trajectory-based whale optimization algorithm for global optimization. *Ieee Access, 5*, 6168–6186.

74. Chen, H. L., Yang, C. J., Heidari, A. A., & Zhao, X. H. (2020). An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Systems with Applications, 154*, 113018.

75. Wei, Y., Lv, H. J., Chen, M. X., Wang, M. J., Heidari, A. A., Chen, H. L., & Li, C. Y. (2020). Predicting entrepreneurial intention of students: An extreme learning machine with Gaussian Barebone Harris hawks optimizer. *Ieee Access, 8*, 76841–76855.

76. Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations, 29*, 464–483.

77. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191.

78. Chen, X., Tianfield, H., Mei, C. L., Du, W. L., & Liu, G. H. (2017). Biogeography-based learning particle swarm optimization. *Soft Computing, 21*, 7519–7541.

79. Mostafa Bozorgi, S., & Yazdani, S. (2019). IWOA: An improved whale optimization algorithm for optimization problems. *Journal of Computational Design and Engineering, 6*, 243–259.

80. Zhuang, Y., Chen, S., Jiang, N., Hu, H. (2022). An effective WSS-ENet-based similarity retrieval method of large lung CT image databases. *KSII Transactions on Internet & Information Systems, 16*, 2359–2376.

81. Zhuang, Y., Jiang, N., & Xu, Y. M. (2022). Progressive distributed and parallel similarity retrieval of large CT image sequences in mobile telemedicine networks. *Wireless Communications & Mobile Computing, 2022*, 13.

82. Lu, S. Y., Yang, B., Xiao, Y., Liu, S., Liu, M. Z., Yin, L. R., & Zheng, W. F. (2023). Iterative reconstruction of low-dose CT based on differential sparse. *Biomedical Signal Processing and Control, 79*, 104204.

83. Qin, X. M., Ban, Y. X., Wu, P., Yang, B., Liu, S., Yin, L. R., Liu, M. Z., & Zheng, W. F. (2022). Improved image fusion method based on sparse decomposition. *Electronics, 11*, 2321.

84. Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters, 44*, 800-U835.

85. Zhang, L., Zhang, L., Mou, X. Q., & Zhang, D. (2011). FSIM: A Feature Similarity Index for Image Quality Assessment. *Ieee Transactions on Image Processing, 20*, 2378–2386.

86. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *Ieee Transactions on Image Processing, 13*, 600–612.

87. Pare, S., Kumar, A., Bajaj, V., & Singh, G. K. (2016). A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve. *Applied Soft Computing, 47*, 76–102.

88. Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging, 13*, 146–168.

89. Agrawal, S., Panda, R., Choudhury, P., & Abraham, A. (2022). Dominant color component and adaptive whale optimization algorithm for multilevel thresholding of color images. *Knowledge-Based Systems., 240*, 108172.

90. Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (vol. 2, pp. 60–65) San Diego, CA, USA.

91. Abutaleb, A. S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing, 47*, 22–32.

92. Liu, L., Zhao, D., Yu, F., Heidari, A. A., Ru, J., Chen, H., Mafarja, M., Turabieh, H., & Pan, Z. (2021). Performance optimization of differential evolution with slime mould algorithm for multilevel breast cancer image segmentation. *Computers in Biology and Medicine, 138*, 104910.

93. Kim, K., Choi, J., & Lee, Y. (2020). Effectiveness of non-local means algorithm with an industrial 3 MeV linac high-energy X-ray system for non-destructive testing. *Sensors, 20*, 2634.

94. Zhao, S., Wang, P., Heidari, A. A., Chen, H., Turabieh, H., Mafarja, M., & Li, C. (2021). Multilevel threshold image segmentation with diffusion association slime mould algorithm and Renyi's entropy for chronic obstructive pulmonary disease. *Computers in Biology and Medicine, 134*, 104427.

95. Liu, W., Huang, Y. K., Ye, Z. W., Cai, W. C., Yang, S., Cheng, X. C., & Frank, I. (2020). Renyi's entropy based multilevel thresholding using a novel meta-heuristics algorithm. *Applied Sciences-Basel, 10*, 3225.

96. Golshani, L., Pasha, E., & Yari, G. (2009). Some properties of Renyi entropy and Renyi entropy rate. *Information Sciences, 179*, 2426–2433.

97. Rényi, A. (1985). On measures of entropy and information. *Virology, 142*, 158–174.

98. Bovik, H. (2006). Image information and visual quality. *Ieee Transactions on Image Processing a Publication of the Ieee Signal Processing Society, 15*, 430.

99. Yim, C., & Bovik, A. C. (2011). Quality assessment of deblocked images. *IEEE Transactions on Image Processing, 20*, 88–98.