Multimedia Tools and Applications (2020) 79:12007–12040
https://doi.org/10.1007/s11042-019-08566-1

An improved thermal exchange optimization based GLCM for multi-level image segmentation

Zhikai Xing[1] · Heming Jia[1]

Received: 12 February 2019 / Revised: 5 November 2019 / Accepted: 6 December 2019 /
Published online: 10 January 2020

Abstract
The gray-level co-occurrence matrix (GLCM) can obtain the pixel matrix of the image, and selecting multiple thresholds for the matrix can obtain better segmentation results. However, as the number of threshold increases, the computational complexity of the algorithm will also increase. In order to solve this problem, this paper proposes a multi-threshold image segmentation method based on thermal exchange optimization (TEO) algorithm, and take a novel diagonal class entropy (DCE) as the fitness function. We improve TEO algorithm by using two strategic methods of Levy flight (LF) and opposition-based learning (OBL). In order to verify the segmentation ability of the proposed algorithm, color natural images, satellite images and Berkeley images are taken as experimental objects to analyze the segmentation result graph and image segmentation quality evaluation indexes. Experimental results show that the GLCM-ITEO algorithm has good segmentation capability, less CPU time.

Keywords Color image segmentation · GLCM · Thermal exchange optimization · Levy flight · Opposition-based learning

1 Introduction

Image segmentation is the basic work of image processing research. There are primarily four types of segmentation methods: thresholding [26, 28, 45], boundary-based [13, 18], region-based [8, 12, 37], and hybrid techniques [7, 9, 20, 30]. The threshold method involves selecting a set of thresholds using some of the characteristics defined from the image. The concept of gray-level co-occurrence matrix (GLCM) considers the spatial correlation among the gray level of image [4, 32]. GLCM has attracted more and more attention, and high quality segmentation images can be obtained by using gray co-occurrence matrix for image

✉ Heming Jia
jiaheming@nefu.edu.cn

[1] College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

segmentation. GLCM is used in many fields, Li M proposed a method on marble texture image segmentation based on Gray Level Co-occurrence Matrix (GLCM) [27]. Zhao M proposed a method of declaring an image based on rich edge region extraction using a gray-level co-occurrence matrix [55]. This method could better extract the edge information of the image. Qayyum R presented a novel approach for classification of wood knot defects for an automated inspection [40]. The proposed technique utilized gray level co-occurrence matrix based features and a particle swarm optimization trained feedforward neural network. The improved GLCM algorithm can improve the segmentation accuracy, so in order to better improve the image segmentation accuracy of the algorithm, it has become a common method to use the optimization algorithm to find the optimal segmentation threshold of the multi-threshold algorithm [21, 31, 51].

The optimization algorithm can solve practical engineering problems in recent years. Different optimization algorithms adapt to different engineering problems and have different optimization capabilities [6, 33]. These nature-inspired optimization algorithms are mainly classified into two classes recently which are evolutionary algorithm (EA) and biology-inspired or bio-inspired algorithms. EA imitated the Darwinian theory of evolution [14]. There were many good algorithms in this class. In 1975, GA was invented by John Holland [16], it used the binary representation of individuals. In 1997, Differential evolution was proposed Rainer stone, the essence was a multi - objective optimization algorithm [46]. The most popular class was the biology-inspired or bio-inspired algorithms right now. One of the most famous algorithm was the Particle Swarm Optimizer (PSO) which was developed based on the swarming behavior of fish and birds [24]. In 2015, the ant lion optimizer was proposed by Mirjalili [35]. In 2016, Askarzadeh proposed the crow search algorithm [3]. In 2017, the killer whale algorithm was proposed by Biyanto [5]. These algorithms were inspired from the predation behavior animal, so as to obtain better searching ability. There also some algorithms inspired from the physics and chemistry, these algorithms usually had simple mathematical models, but had good optimization effect. In 2001, the harmony search algorithm was proposed by Geem [15]. In 2015, Zheng Yu-Jun proposed water wave optimization algorithm [56]. In 2017, Kaveh A proposed the thermal exchange optimization [23]. The algorithm was proposed according to Newton’s law of cooling. Because of the simple mathematical model, these algorithms that imitated physical phenomena were easy to fall into local optimum when solving complex optimization problems. So, using different strategies to improve the optimization algorithm can better solve engineering problems.

Although there is no perfect optimization algorithm, the optimization algorithm can be improved to make it more suitable for solving engineering problems. The strategy commonly used by scholars was Levy-flight. Levy flight (LF) was a random walk strategy whose step length obeyed the Levy distribution [29, 48, 57]. Yan Bailu proposed a particle swarm optimization algorithm which used random learning mechanism and levy flight as the improved strategies [52]. The algorithm can be found faster and more efficiently. Mesa A used Levy flight improve the cuckoo search algorithm [34]. The results show that compared with particle swarm optimization and other existing algorithms, the algorithm can obtain better facility location. Mousavirad S J proposed a simple but efficient population-based metaheuristic algorithm called human mental search (HMS) [36]. The algorithm performed a mental search of HMS, exploring the area around each solution based on Levy flight. Heidari A A proposed an improved modified GWO algorithm for solving either global or real-world optimization problems [19]. Opposition-based learning (OBL) as a new scheme for machine intelligence was introduced by Tizhoosh H R [47]. This new approach was based on estimates and counter-estimates, weights and relative weights, actions and counteractions. Ahmed A E proposed an improved version of the grasshopper optimization algorithm (GOA) based on the opposition-based learning (OBL) strategy called OBL-GOA for solving benchmark optimization functions and engineering problems [1]. Experiments show that the results of this algorithm were better than those of ten famous algorithms in this field. Dong W improved particle swarm optimization with OBL [10]. This method solved the problem of premature convergence of traditional particle swarm optimization. So, the strategic methods can increase the jump step length of individual population and increase the search range of individual population.

In this paper, we mainly study the threshold selection of multi-threshold GLCM. As the number of thresholds increases, the operation time of the algorithm increases and the segmentation accuracy decreases. We propose an improved TEO algorithm to optimize the multi-threshold GLCM algorithm, and use LF and OBL to improve the TEO algorithm to improve its optimization ability. The DCE is used as the fitness function to find the image threshold more accurate. The ITEO algorithm can effectively improve the segmentation accuracy of multi-threshold GLCM and reduce the overall running time of the algorithm.

2 Material and methods

2.1 Problem assessment of multilevel thresholding

The process of searching optimal thresholding values of a given image is considered as a constrained optimization problem. For bi-level thresholding, the problem is to find an optimal valueT^*. If the image intensity $I_{i,j}$ is less than the value T^*, the pixel in an image is replaced with a black pixel or a white pixel if the image intensity is greater than that constant T^*, the expression can be stated as follows:

$$g(x,y) = \begin{cases} 1 \quad if \quad f(x,y) > T^* \\ 0 \quad if \quad f(x,y) < T^* \end{cases} \tag{1}$$

The problem can be extended to multilevel thresholding that has more than one threshold value and divide the original image into multiple classes:

$$\begin{aligned} N_0 &= \{g(x,y) \in I | 0 \le g(x,y) \le t_1 - 1|\} \\ N_1 &= \{g(x,y) \in I | t_1 \le g(x,y) \le t_2 - 1|\} \\ N_i &= \{g(x,y) \in I | t_i \le g(x,y) \le t_i - 1|\} \\ N_n &= \{g(x,y) \in I | t_n \le g(x,y) \le L - 1|\} \end{aligned} \tag{2}$$

where N_i is the ith class, n is the number of threshold values, and $t_i (i = 1, \cdots, n)$ is the ith threshold value.

2.2 Grey-level co-occurrence matrix (GLCM)

GLCM is a second-order statistical method that computes the frequency of pixel pairs having same gray-levels in an image and applies additional knowledge obtained using spatial pixel relations [22, 38]. Co-occurrence matrix embeds distribution of grayscale transitions using edge information. Since, most of the information required for computing threshold values is embedded in GLCM, it has emerged as a simple yet effective technique.

Consider I as an image with 0 to L quantized gray-levels, L is considered as 256. Each matrix element of the GLCM contains the second-order statistics, probability values for changes between gray levels i and j for a particular displacement and angle. For a given distance, four angular GLCM are defined for θ = 0°, 45°, 90°, and 135°.

$$G = [g(d, 0^{\circ}) + g(d, 45^{\circ}) + g(d, 90^{\circ}) + g(d, 135^{\circ})]/4 \tag{3}$$

Where $g(\bullet)$ denotes GLCM in one direction only. Next, to prevent a negative value occurring for the entropy, we normalize the final GLCM as:

$$G(i,j) = g(i,j) / \sum_{i=1}^{L} \sum_{j=1}^{L} g(i,j) \tag{4}$$

In this paper, we use the entropy feature computed from the GLCM. Let L be the number of gray levels in the image. Then the size of GLCM will be L × L. Let G(i, j) represents an element of the matrix. Then the entropy feature from the matrix is computed as

$$H = -\sum_{i=1}^{L} \sum_{i=1}^{L} G(i,j) \times \ln(G(i,j)) \tag{5}$$

However, for bi-level thresholding, for a threshold value T, the DCE is computed as

$$H_A = -\sum_{i=1}^{T} \sum_{i=1}^{T} G(i,j) \times \ln(G(i,j)) \tag{6}$$

$$H_C = -\sum_{i=T+1}^{L} \sum_{i=T+1}^{L} G(i,j) \times \ln(G(i,j)) \tag{7}$$

$$H_{DCE}(T) = H_A(T) + H_C(T) \tag{8}$$

When this formulation is extended to multilevel thresholding, we consider only the diagonal regions of the GLCM for computing the DCE for each level of thresholding. The optimum thresholds are obtained when DCE is minimized. We introduce here the theoretical formulation for multilevel thresholding using DCE. For (K–1) thresholds $[T_1, T_2, ..., T_{K-1}]$ the DCE is computed as

$$\begin{aligned} H_{DCE}(T_1, T_2, ..., T_{K-1}) = &-\sum_{i=1}^{T_1} \sum_{i=1}^{T_1} G(i,j) \times \ln(G(i,j)) - \sum_{i=T_1}^{T_2} \sum_{i=T_1}^{T_2} G(i,j) \\ &\times \ln(G(i,j)) \cdots - \sum_{i=T_{K-1}+1}^{L} \sum_{i=T_{K-1}+1}^{L} G(i,j) \times \ln(G(i,j)) \end{aligned} \tag{9}$$

The proposed objective function is:

$$\{T_1, T_2, ..., T_{K-1}\} = \operatorname{argmin}\{H_{DCE}(T_1, T_2, ..., T_{K-1})\} \tag{10}$$

Where, K is the number of classes.

2.3 Between-class variance method (Otsu's method)

The Otsu based between-class variance method has been employed in determining the optimal thresholding values of an image. The Otsu's method can be described as follows: assume that an image can be represented in L gray levels (1,2, …,L) and has N pixels [25]. The number of pixels at level i are denoted byf_1, and $N = f_1 + f_2 + \cdots f_i$.Then, the occurrence probability of gray level i can be defined by the following equation:

$$p_i = \frac{f_i}{N}, p_i \geq 0, \sum_{i=1}^{L} p_i = 1 \tag{11}$$

In bi-level thresholding, the optimum threshold t divides the image into two classes, and the cumulative probabilities of each class can be described as follows:

$$\varpi_0 = \sum_{i=1}^{t} p_i, \varpi_1 = \sum_{i=t+1}^{L} p_i \tag{12}$$

The mean levels of two classes can be defined as follows:

$$\mu_0 = \sum_{i=1}^{t} ip_i/\varpi_0, \mu_1 = \sum_{i=t+1}^{L} ip_i/\varpi_1 \tag{13}$$

Let μ_T be the mean levels of the whole image and it can be defined by

$$\mu_T = \sum_{i=1}^{L} ip_i \tag{14}$$

The between-class variance of whole classes can be represented by

$$f(t) = \sigma_0 + \sigma_1 \tag{15}$$

Where$\sigma_0 = \varpi_0(\mu_0 - \mu_T)^2and\sigma_1 = \varpi_1(\mu_1 - \mu_T)^2$. For bi-level thresholding, the Otsu's method find an optimal threshold $t^*$by maximizing the between-class variance, that is:

$$t^* = \text{argmax}(f(t)) \tag{16}$$

The Otsu's method can be also extended to multi-level thresholding. Assuming that there are m thresholds, which divide the image into $m + 1$ classes. The extended between-class variance is calculated by

$$f(t) = \sum_{i=0}^{m} \sigma_i \tag{17}$$

The sigma terms are determined by Eq. 18 and the mean levels are calculated by Eq. 19:

$$\begin{gathered} \sigma_0 = \varpi_0(\mu_0 - \mu_T)^2, \sigma_1 = \varpi_1(\mu_1 - \mu_T)^2, \cdots, \\ \sigma_{M-1} = \varpi_{M-1}(\mu_{M-1} - \mu_T)^2 \end{gathered} \tag{18}$$

$$\begin{gathered} \mu_0 = \sum_{i=1}^{t} ip_i/\varpi_0, \mu_1 = \sum_{i=t_1+1}^{t_2} ip_i/\varpi_1, \cdots, \\ \mu_{M-1} = \sum_{i=t_{M-1}+1}^{L} ip_i/\varpi_{M-1} \end{gathered} \tag{19}$$

Fig. 1 Hot iron objects, transferring heat to the surrounding environment

The optimum thresholds are found by maximizing the between-class variance by Eq. 20:

$$t^{*} = \operatorname{argmax}\left(\sum_{i=0}^{M-1} \sigma_i\right) \tag{20}$$

2.4 Thermal exchange optimization

The TEO is a new optimization algorithm based on Newton's law of cooling which the rate of heat loss of a body is proportional to the difference in temperatures between the body and its

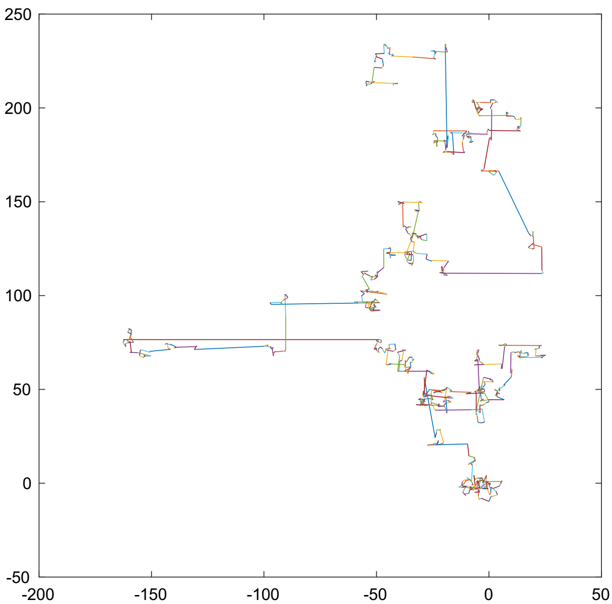


Fig. 2 Levy's flight path

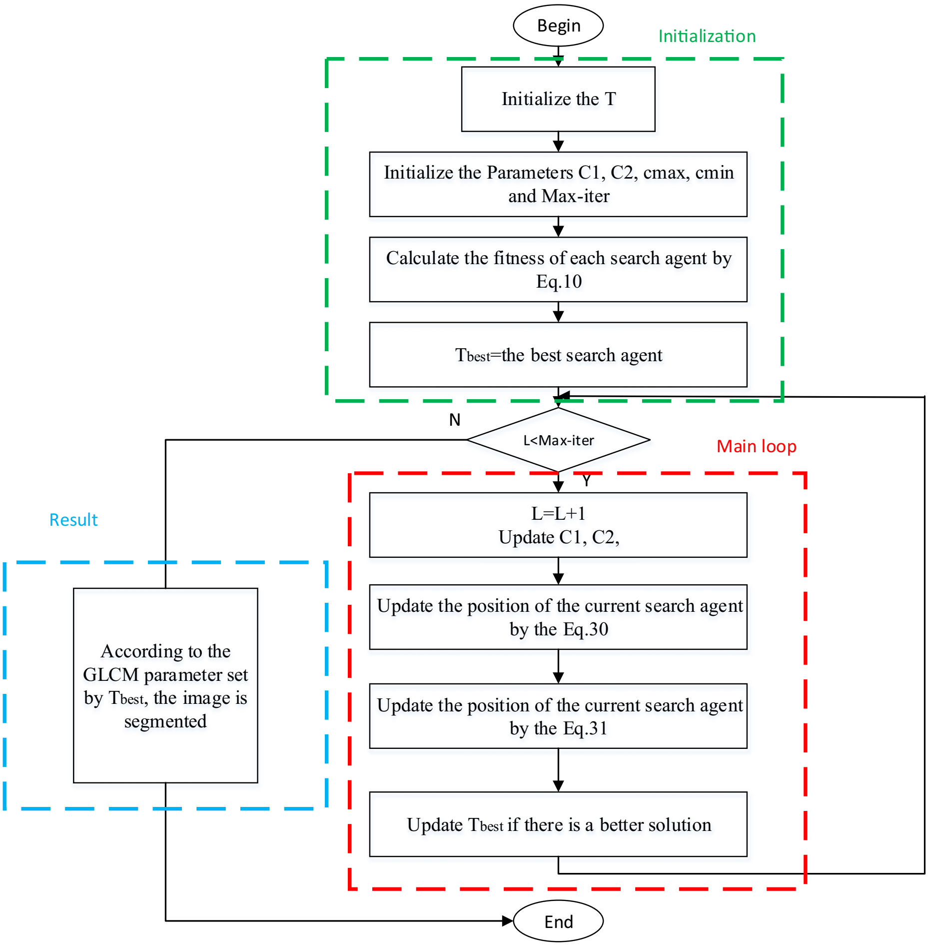
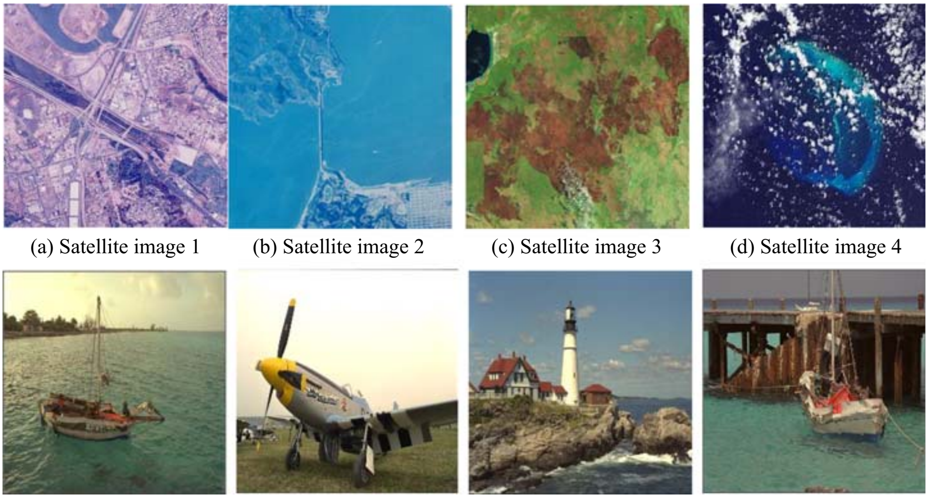


Fig. 3 The flow chart of the ITEO-GLCM

surroundings. The hot iron objects transferring heat to the surrounding environment is shown in Fig. 1.

Table 1 Parameters and references of the comparison algorithms

Algorithm	Parameters	Value
TEO	c_1	2
	c_2	2
CSA [39]	AP	0.5
FPA [50]	P	0.5
PSO [49]	Swam size	200
	Cognitive, social acceleration	2,2
	Inertial weight	0.95–0.4
BA [53]	β	(0,1)
ITEO	Levy	1.5



(a) Satellite image 1 (b) Satellite image 2 (c) Satellite image 3 (d) Satellite image 4

(e) Kodim image 1 (f) Kodim image 2 (g) Kodim image 3 (h) Kodim image 4

Fig. 4 The color test images

In TEO algorithm, some agents are defined as the cooling objects and the remaining agents are supposed to represent the environment. Updating the temperature formula between objects can be defined as:

$$T_i^{env} = (1-(c_1 + c_2 \times (1-t)) \times random) \times T_i^{'env} \tag{21}$$

$$t = \frac{l}{L} \tag{22}$$

where c_1, c_2 are the controlling variables, $T_i^{'env}$ is the previous temperature of the object, which is modified to T_i^{env}. l is the current iteration number, L is the max iteration number.

According to the previous steps and Eq. 23, new temperature of each object is updated by

$$T_i^{new} = T_i^{env} + \left(T_i^{old} - T_i^{env}\right)\exp(-\beta t) \tag{23}$$

$$\beta = \frac{\mathrm{Cost}(object)}{\mathrm{Cost}(worst-object)} \tag{24}$$

Table 2 The AP value of ITEO under different parameters

Function	ITEO(β = 0.5)	ITEO(β = 1)	ITEO(β = 1.5)	ITEO(β = 2)	TEO
Satellite image1	0.6777	0.7102	**0.7125**	0.6902	0.6105
Satellite image2	0.6580	0.7057	**0.7192**	0.6847	0.6037
Satellite image3	0.6395	0.7002	**0.7226**	0.6825	0.5951
Satellite image4	0.6163	0.6840	**0.7389**	0.6027	0.5765
Kodim image1	0.6157	0.6745	**0.7437**	0.6014	0.5758
Kodim image2	0.6388	0.6943	**0.7251**	0.6736	0.5900
Kodim image3	0.6224	0.7001	**0.7324**	0.6308	0.5879
Kodim image4	0.6190	0.6930	**0.7375**	0.6220	0.5836

Table 3 The IoU value of ITEO under different parameters

Function	ITEO(β = 0.5)	ITEO(β = 1)	ITEO(β = 1.5)	ITEO(β = 2)	TEO
Satellite image1	0.5306	0.5837	**0.5933**	0.5404	0.5249
Satellite image2	0.5006	0.5947	**0.6143**	0.5284	0.5058
Satellite image3	0.5816	0.5506	**0.6243**	0.4951	0.4998
Satellite image4	0.5778	0.5186	**0.6411**	0.5391	0.4679
Kodim image1	0.4667	0.5516	**0.5715**	0.5604	0.4222
Kodim image2	0.5706	0.5201	**0.5867**	0.5114	0.5428
Kodim image3	0.4801	0.5370	**0.6160**	0.5097	0.5557
Kodim image4	0.5749	0.5067	**0.6052**	0.4808	0.4691

Table 4 The AP and IOU of each algorithm under GLCM

T	CSA		FPA		PSO		BA		ITEO	
	AP	IoU	AP	IoU	AP	IoU	AP	IoU	AP	IoU
Satellite image1										
4	0.7069	0.6177	0.6941	0.5032	0.6105	0.5757	0.6869	0.6043	**0.7125**	**0.6472**
6	0.7145	0.6184	0.6951	0.5113	0.6138	0.5846	0.6909	0.6132	**0.7224**	**0.6542**
8	0.7192	0.6220	0.7017	0.5183	0.6165	0.5944	0.6992	0.6157	**0.7229**	**0.6565**
12	0.7289	0.6270	0.7041	0.5256	0.6183	0.5974	0.7029	0.6237	**0.7280**	**0.6591**
Satellite image2										
4	0.7115	0.6237	0.6819	0.5676	0.6034	0.5268	0.6765	0.5932	**0.7127**	**0.6775**
6	0.7161	0.6288	0.6899	0.5680	0.6101	0.5313	0.6849	0.6027	**0.7167**	**0.6846**
8	0.7227	0.6344	0.6922	0.5691	0.6189	0.5381	0.6893	0.6099	**0.7204**	**0.6906**
12	0.7298	0.6381	0.7010	0.5726	0.6257	0.5384	0.6969	0.6170	**0.7208**	**0.6974**
Satellite image3										
4	0.7091	0.5305	0.6679	0.5016	0.5965	0.4923	0.6747	0.5806	**0.7190**	**0.6260**
6	0.7162	0.5312	0.6682	0.5062	0.6007	0.4983	0.6757	0.5809	**0.7283**	**0.6271**
8	0.7180	0.5386	0.6748	0.5150	0.6035	0.5076	0.6778	0.5821	**0.7329**	**0.6323**
12	0.7258	0.5401	0.6825	0.5211	0.6040	0.5117	0.6783	0.5846	**0.7414**	**0.6329**
Satellite image4										
4	0.7050	0.5336	0.6500	0.5411	0.5916	0.5178	0.6717	0.5510	**0.7236**	**0.6388**
6	0.7116	0.5391	0.6568	0.5511	0.5940	0.5231	0.6786	0.5592	**0.7238**	**0.6417**
8	0.7185	0.5413	0.6634	0.5511	0.5959	0.5275	0.6878	0.5665	**0.7328**	**0.6460**
12	0.7209	0.5497	0.6701	0.5587	0.5983	0.5334	0.6911	0.5675	**0.7424**	**0.6559**
Kodim image1										
4	0.7037	0.5820	0.6366	0.4842	0.5819	0.4513	0.6570	0.5361	**0.7275**	**0.6012**
6	0.7083	0.5919	0.6447	0.4926	0.5867	0.4527	0.6592	0.5411	**0.7321**	**0.6081**
8	0.7092	0.5953	0.6501	0.4980	0.5957	0.4543	0.6691	0.5475	**0.7399**	**0.6167**
12	0.7119	0.6010	0.6568	0.5038	0.5981	0.4606	0.6773	0.5514	**0.7459**	**0.6223**
Kodim image2										
4	0.7003	0.5816	0.6340	0.4977	0.5742	0.5818	0.6566	0.5796	**0.7355**	**0.6101**
6	0.7102	0.5913	0.6381	0.5004	0.5763	0.5830	0.6584	0.5819	**0.7440**	**0.6176**
8	0.7104	0.6003	0.6475	0.5077	0.5806	0.5850	0.6667	0.5884	**0.7506**	**0.6201**
12	0.7126	0.6016	0.6515	0.5087	0.5830	0.5918	0.6716	0.5973	**0.7524**	**0.6222**
Kodim image3										
4	0.6948	0.6050	0.6309	0.4873	0.5726	0.4575	0.6393	0.5478	**0.7438**	**0.6198**
6	0.7030	0.6096	0.6311	0.4898	0.5820	0.4640	0.6460	0.5549	**0.7479**	**0.6245**
8	0.7056	0.6145	0.6373	0.4952	0.5821	0.4682	0.6467	0.5605	**0.7565**	**0.6246**
12	0.7065	0.6190	0.6376	0.4984	0.5823	0.4716	0.6500	0.5629	**0.7623**	**0.6260**
Kodim image4										
4	0.6934	0.5737	0.6214	0.5516	0.5637	0.4534	0.6218	0.5880	**0.7521**	**0.6489**
6	0.6992	0.5809	0.6273	0.5596	0.5648	0.4596	0.6310	0.5930	**0.7540**	**0.6515**
8	0.7035	0.5900	0.6325	0.5597	0.5671	0.4632	0.6331	0.5985	**0.7588**	**0.6516**
12	0.7066	0.5901	0.6386	0.5599	0.5714	0.4638	0.6412	0.6021	**0.7600**	**0.6600**

Table 5 The threshold levels of each algorithm under GLCM

T	CSA			FPA	
	R	G	B	R	G
Satellite image1					
4	82,113,166,190	45 91,143,175	105,114,134,165	71 93,157,193	65,107,134,171
6	74,100,120,159,177,231	7 43 79,111,138,183	66,106,128,157,174,196	40 78,116,148,188,217	29 77,106,109,152,188
8	47 80 84,127,138,170, 188,219	46 47 82 93,136,172, 208,230	84,112,112,140,151, 166,176,214	35 78 90,105,136, 154,184,210	18 47 78 92,123,142,165,192
12	1 44 82,101,119,121,145, 147,177,193,208,219	31 45 60 78 96 98,106,112, 137,147,171,192	33 79,111,132,149,165, 166,172,182,200,202,233		
Satellite image2				24 43 65 87	54 58,113,153
4	38 72 97,137	121,137,146,215	93,149,158,212	21 40 53 68,127,171	101,120,140,141,183,210
6	26 56,104,135,159,182	95,124,147,151,158,191	30 68,108,138,160,178	19 40 56 69 77,108, 158,177	29 97,127,141,159,173,206,221
8	15 15 36 46 66 90, 117,185	58 80,104,113,130, 150,192,194	58,116,136,136,145, 163,193,208	7 21 32 36 38 41 49 62 78,140,186,208	46 66 95,101,103,115, 122,137,151,172,190,207
12	15 16 25 37 56 84 97, 113,165,205, 216,231	65 66 77,107,108,120, 125,130,139,152,157,185	71 77 93,122,132,148,150, 164,167,185,187,187		
Satellite image3				53 83,123,153	71,101,140,175
4	84,124,128,148	56,100,141,179	27 51 65,111	61 72,103,133,162,176	46 71,103,112,141,188
6	15 84,126,145,163,191	17 57,107,122,150,177	24 40 57 66 73,180	61 89,101,134,155, 170,171,187	42 69 81 97,130, 154,173,174
8	70 91,101,119,130,158,180,186	36 57 77,113,135,162, 221,256	17 28 32 39 66 73 92,135		
12	68 71 95,106,112,122,122, 133,141,147,178,183	36 40 42 66 88,107,124,144, 160,166,175,177	3 15 22 28 43 62 77 79 93,108,126,127	27 70,129,180	51 68,121,193
Satellite image4				14 27 59 83 93,152	41 67 91,104,158,159
4	25 39,102,124	48 76,128,178	50 79,107,122	21 26 29 39 92,153, 171,194	38 60 65 82 98, 122,135,180
6	18 28 47 52,118,166	18 43 70 86,113,159	38 48 74 96,144,154		
8	25 32 44 51 53 55 92,157	24 39 63 84,104,145, 157,189	4 11 42 58 74 85 91,144	54 86,180,223	83,129,179,245
12				56,107,123,164,188,214	67,113,114,165,185,227

Table 5 (continued)

T	CSA			FPA	
	19 27 36 69 79,115,155,157,168, 183,184,203	38 46 57 66 78 92,108,130, 153,204,207,214	9 13 13 18 48 52 56 62 77 93,142,195		
Kodim image1				57 80 94,121,155, 212,221,243	75 93,123,125,147, 196,227,233
4	46 84,138,171	82 94,129,190	67,114,149,188		
6	50 95,112,142,177,222	37 80,101,154,196,229	51 94,117,120,164,193	73 94,118,159	55 98,132,177
8	15 59 79 91,141,178,194,239	36 83 89,126,139, 180,197,223	40 58 61 69,114,139,167,206	41 50 79,117,135,182	41 56 89 94,129,165
12	6 13 54 80 97,102,118,136,170, 184,193,239	17 74 89,112,119,135, 158,168,187,203, 232,240	49 74 82 90 95,113,129, 158,171,191,203,217	44 47 84,102,114, 126,143,203	41 48 82 97,130, 138,160,197
Kodim image2					
4	58 91,126,170	74,106,138,178	69 84,131,153	34 95,134,153	94,134,148,167
6	67 75,103,139,158,165	57 68 98,113,152,196	45 54 73 96,130,154	63 99,113,118,136,184	58 84,121,144,151,206
8	43 74 75,103,115,131, 141,195	42 47 64 69 70,103,125,165	25 40 54 89 90,130,139,191	5 59 79,104,119,140, 179,250	48 86 88,117,143,171, 203,252
12	39 56 80 82 86 95,104,127, 162,175,209,209	9 25 44 66 69 97,124,135, 153,186,219,238	11 37 39 60 78 99,126, 154,198,198,209,247		
Kodim image3				52 91,126,191	38 83,125,194
4	67 98,116,159	66 93,115,159	36 94,129,164	61 84,115,165,170,203	9 42 66,114,166,211
6	48 77,107,116,148,189	25 46 99,130,149,176	25 60,103,140,159,195	19 59 71 89,124, 126,157,216	40 73 97,112,149, 170,183,186
8	40 61 92,114,140,185,256,256	66 88,116,133,133, 146,159,195	40 83,135,144,156, 169,218,235	71 93,157,193	65,107,134,171
12	19 21 40 68 93,103,112,116, 137,151,188,191	3 18 29 35 59 77 90, 123,142,153,209,239	4 9 25 57 73 87 97,141, 155,160,186,201	40 78,116,148,188,217	29 77,106,109,152,188
Kodim image4				35 78 90,105,136, 154,184,210	18 47 78 92,123, 142,165,192
4	55,101,153,196	46 98,141,173	36 62 85,193		
6	58 84,105,135,177,197	45 85,109,136,145,189	19 45 63 95,142,185	24 43 65 87	54 58,113,153
8	59 88,103,137,147,177, 180,202	37 49 86,104,131, 160,191,204	18 36 53 72 90,120, 154,200	21 40 53 68,127,171	101,120,140,141,183,210

Table 5 (continued)

T	CSA			FPA	
12	6 39 51 72 94,109,110,123, 138,148,191,219	5 43 45 70 95,117,124,154, 159,183,202,218	29 37 58 62 63 73 81, 119,156,183,218,238	19 40 56 69 77,108, 158,177	29 97,127,141,159, 173,206,221

T		PSO		
	B	R	G	B
Satellite image1				
4	66 92,144,176	40,116,150,197	53 91,134,170	71,110,159,179
6	48,113,136,153,190,252	63,101,124,144,154,203	67 95,100,143,161,202	91,114,136,146,151,176
8	75 97,110,124,133,162,165,185	34 79 94,123,158,188,198,217	55 64 97,112,146,161,197,203	56 93,132,141,149,162,167,193
12				
Satellite image2	12,132,149,177	37 61,166,197	107,120,145,200	110,155,173,197
4	70 84 94,135,158,172	33 48 67 70 86,146	113,114,135,150,152,198	86,112,133,151,163,211
6	9 57 71,120,134,152,175,207	17 46 48 57 88,108,166,191	1 25 84,112,132,141,145,163	115,148,149,154,166,184,210,229
8	13 34 81 95,101,118,144, 155,160,173,196,227	29 37 59 65 75 79,113,123,143,183,192,212	4 50 78,114,117,126,131,138, 138,153,159,195	77 77 84 97,119,153,164,174, 195,213,241,243
12				
Satellite image3	24 36 67,122	70,100,129,175	58 99,144,163	33 39 78,143
4	27 45 65 89,108,217	66 85,101,124,145,172	12 55 94,141,148,173	26 42 58 67,109,122
6	28 34 42 48 60 99,108,117	53 81,101,124,133,151,196,220	42 60 76,100,133,147,190,195	6 29 41 62 70 86 95,143
8				
12	39 61 88,143	24 49 92,164	59 90,104,125	17 53 75,111
Satellite image4	37 55 83,113,159,206	20 26 45 60 91,119	33 51 77,109,127,202	45 57 73,106,172,211
4	25 50 55 74,100,134,199,200	14 25 30 39 55 83,135,200	31 32 52 68 93,154,173,194	9 42 57 66 84,104,172,222
6				
8	55 97,114,159	68,101,144,202	60,116,144,187	86,121,163,185
12	63 88,100,142,192,197	50 70,114,139,190,196	51,104,138,143,172,207	51 71 96,143,180,205
Kodim image1	42 49 61 87,109,150,191,207	43 75 99,132,140,161,175,217	54 77 89,110,122,130,152,216	25 65 86 97,155,173,175,199
4				
6	41 96,147,157	55 77,117,147	66 95,130,149	76,131,178,186
8	31 60 96,116,122,154	36 63 90,113,155,188	39 63,108,145,178,231	50 69 96,137,144,164
12	27 49 83,111,125,141,189,222	48 75 76 91 96,123,155,210	52 71 93,121,136,158,200,205	40 65 69 96,115,139,167,174

Table 5 (continued)

T		PSO		
Kodim image2				
4	47 89,118,163	71,113,132,175	62 87,143,162	50 68,118,162
6	52 65,131,146,165,247	68 87 98,125,141,232	79 95,113,142,148,180	30 64,106,143,161,217
8	33 55,101,128,150,160,212,243	49 64 88,108,132,143,153,230	8 52 75 90,104,106,138,165	35 55 67,121,123,142,161,227
12				
Kodim image3	59 72,122,167	64,105,181,204	52 83,127,183	36 55 95,171
4	35 55 63 93,126,185	59 79,106,124,147,205	50 82,123,144,200,205	27 48 77,115,152,158
6	20 49 52 76,107,143,167,194	19 45 71 91,101,113,174,214	31 52 65 84 89,112,153,183	35 54 74 84,142,151,183,214
8	66 92,144,176	40,116,150,197	53 91,134,170	71,110,159,179
12	48,113,136,153,190,252	63,101,124,144,154,203	67 95,100,143,161,202	91,114,136,146,151,176
Kodim image4	75 97,110,124,133,162,165,185	34 79 94,123,158,188,198,217	55 64 97,112,146,161,197,203	56 93,132,141,149,162,167,193
4				
6	12,132,149,177	37 61,166,197	107,120,145,200	110,155,173,197
8	70 84 94,135,158,172	33 48 67 70 86,146	113,114,135,150,152,198	86,112,133,151,163,211
12	9 57 71,120,134,152,175,207	17 46 48 57 88,108,166,191	1 25 84,112,132,141,145,163	115,148,149,154,166,184,210,229

Table 6 The optimal fitness value of each algorithm under GLCM

T	BA			ITEO		
	R	G	B	R	G	B
Satellite image1						
4	95,135,170,192	71,113,128,194	92,133,159,239	58,103,155,222	61 90,127,190	85,123,159,199
6	45 68,104,142,185, 199	43 80 87,122,162,194	66 83,127,131, 149,195	46 52 92,134, 153,198	45 77,101,122,164,180	19 87,117,147,183,213
8	40 70 89,109,115, 139,184,229	25 60 89,114, 116,123,157,216	79,102,123,142, 184,191,202,215	56 91,114,122, 147,159,181,227	41 78,102,116, 146,152,194,195	32 85,122,134,146, 157,185,220
12						
Satellite image2	36 59,122,169	48 56,118,147	74,120,156,169	6 38 65,144	71,122,136,153	123,160,171,181
4	12 44 75,124, 171,194	103,131,132, 136,145,235	101,132,137,160, 189,197	31 45 75,143, 177,185	67 99,107,133,142,176	132,144,156,169,179,183
6	15 19 37 61 68 93,138,199	95,115,123,137, 144,172,218,236	52 85,106,145, 152,169,201,217	19 42 44 53 71 82,143,170	47 82,110,111,133, 148,214,233	3 6 35,141,159,177,184,209
8	17 26 28 33 48 56 88 92,105,113, 165,217	2 16 47 76,101, 123,132,134,145, 171,179,240	2 22 85 90 96,130, 149,162,166, 177,238,254	22 33 42 47 57 72,122,134,149, 181,205,230	42 99,108,122, 124,140,150,163, 164,183,198,251	50 65 79,131,131,149, 161,171,184,213,224,256
12						
Satellite image3	46 97,122,186	74 98,137,150	39 53 65,119	57 99,128,167	69 89,124,176	33 51 66 98
4	79,110,115,134, 164,206	58 84 91,119 143 176	28 37 56 89, 145,216	81,105,107,129, 141,169	19 52 79,120,136,176	7 33 42 61,119,130
6	48 52 73 96,113, 142,165,171	47 69 85,104,117, 145,152,185	21 35 50 62 87, 116,152,156	48 71 90,111, 123,146, 157,186	12 62 90,114, 116,121,139,174	32 44 57 74 93 94,157,173
8						
12	27 59 82,132	51 83,131,145	41 72 91,130	26 48 84,116	41 73,128,142	56 68 85,152
Satellite image4	22 35 70,141, 187,221	46 62 92,128,142,198	20 49 62 87,107,177	14 33 73 79, 131,169	48 77 80,102,130,143	48 62 62,100,170,214
4	20 40 69,120,136, 150,183,196	48 62 84 88 92 93,129,173	23 38 56 67 86 94,144,170	17 25 34 43 73 87,128,186	11 58 67 92,124, 153,210,250	49 62 65 84,105, 140,148,219
6						
8	58 76,112,201	73,124,179,234	43 76,143,191	41 63,116,192	79,110,154,165	38 89,123,196
12	53 90,116,143, 178,241	46 52 91,126, 169,206	54 81,126,155, 192,236	56 92,106, 136,175,234	55 96,142,171,171,208	66,100,116,144,162,201
Kodim image1	33 45 69,115,155, 185,213,239	13 44 69,117, 138,178,178,230	38 77 95,105,129, 137,175,201	58 73,114, 131,150,172, 196,219	35 70 99,143,163, 196,209,217	39 41 70 98,110, 132,155,194

Table 6 (continued)

T	BA			ITEO		
4						
6	72 79,108,162	49 78,130,155	55 76,103,148	57 98,125,162	67,117,124,165	29 45,131,175
8	23 46 77,111, 148,194	54 97,112,124, 149,215	40 87,105,123, 162,206	25 55 93,110, 150,172	61 65 88,103,140,167	57 90,130,144,150,201
12	29 40 56 91,114, 126,153,186	44 72,105,120, 124,158,186,212	51 55 84,112,135, 162,162,219	47 67 77 91 95,131, 153,175	44 50 51 92,116, 120,156,199	17 35 43 62 86 95,138,166
Kodim image2						
4	56 99,116,157	75,121,153,226	15 74,136,167	9 85,117,181	42 98,137,164	55,109,154,177
6	38 88,118, 128,157,222	5 41 61,102,135,162	48,103,106,131, 156,166	57 79,103, 132,144,164	44 85,112,132,154,184	8 26 67,120,156,162
8	58 84,101,105, 128,155, 227,256	3 52 80 89,112, 126,149,174	53 61 81,116,131, 147,161,190	38 51 70,101, 108,116, 152,165	61 84,105,120,139, 155,158,171	23 38 62 85,144,146,161,165
12						
Kodim image3	61,102,121,220	74,117,171,213	39 71 99,143	54,102,162,205	40 82,103,176	37 75,113,181
4	35 70,108,147, 166,221	27 40 81,106, 136,165	37 61 91,138, 177,178	79,100,119, 132,171,204	24 59,106,136,160,193	36 68 96,147,154,164
6	51 78 89,107,143, 199,210,223	33 56 65 76, 115,124,172,191	29 43 46 81,110, 144,181,221	8 10 70 92, 103,149,178,211	50 62 80,100,141, 174,187,204	31 50 63 94,143,150,160,196
8	95,135,170,192	71,113,128,194	92,133,159,239	58,103,155,222	61 90,127,190	85,123,159,199
12	45 68,104,142, 185,199	43 80 87,122, 162,194	66 83,127,131, 149,195	46 52 92,134,153,198	45 77,101,122,164,180	19 87,117,147,183,213
Kodim image4	40 70 89,109,115, 139,184,229	25 60 89,114, 116,123,157,216	79,102,123,142, 184,191,202,215	56 91,114, 122,147,159, 181,227	41 78,102,116, 146,152,194,195	32 85,122,134, 146,157,185,220
4						
6	36 59,122,169	48 56,118,147	74,120,156,169	6 38 65,144	71,122,136,153	123,160,171,181
8	12 44 75,124, 171,194	103,131,132,136, 145,235	101,132,137, 160,189,197	31 45 75, 143,177,185	67 99,107,133, 142,176	132,144,156,169,179,183
12	15 19 37 61 68 93,138,199	95,115,123,137, 144,172,218,236	52 85,106,145, 152,169,201,217	19 42 44 53 71 82,143,170	47 82,110,111,133, 148,214,233	3 6 35,141,159,177,184,209

Where, the nature when an object has lower β, it exchanges the temperature slightly. The value of β for each object is evaluated according Eq. 24. The $Cost(object)$ is the current value of the target object, and the $Cost(worst-object)$ is the worst value of the target object.

To prevent the temperature of the object from falling into local optimum, set the parameter Pro. It is specified whether a component of each cooling object must be changed or not. If rand<Pro, one dimension of the ith agent is selected randomly and its value is regenerated as follows:

$$T_{i,j} = T_{i,\min} + rand \times \left(T_{j,\max} - T_{j,\min}\right) \tag{25}$$

Where, $T_{i,j}$ is the j th variable of the ith agent. $T_{j,\max}$ and $T_{j,\min}$ are the lower and upper bounds of the j th variable.

The general framework of TEO as follows:

Algorithm 1 TEO

Begin

Initialize the temperature $T_i (i = 1, 2, \ldots, n)$;

Initialize *cmax,cmin,*and *maximum number of iterations;*

Calculate the fitness of each search agent;

Cost=the best search agent ;

While (l<Max number of iterations)

Update t and β

for each search agent

Update the position of the current search agent by the Eq.23 and Eq.25;

end for

Update Cost if there is a better solution;

l= l+1

end while

Return Cost

End

2.5 Levy flight trajectory

Levy's flight is a random step that describes the Levy distribution. Numerous studies have shown that the behavior of many animals and insects is a classic feature of Levy's flight [54]. Levy flight is a special random step method, which is a simulation of the flight path of Levy. The Fig. 2 is the simulation of two-dimensional levy flight by Mantegna method. Its step length is always small, but occasionally it will also appear large pulsation.

The formula for Levy flight is as follows:

$$Levy{\sim}u = t^{-\lambda}, 1 < \lambda \leq 3 \tag{26}$$

The formula for generating Levy random step proposed by Mantegna is as follows:

$$\mathrm{s} = \frac{\mu}{|v|^{1/\beta}} \tag{27}$$

where, parameter β= 1.5, $\mu = \mathrm{N}\left(0, \sigma_{\mu}^{2}\right)$ and $\mathrm{v} = \mathrm{N}\left(0, \sigma_{\mu}^{2}\right)$ are gamma functions.

The variance of the parameters as follows:

$$\sigma_{\mu} = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta \times 2^{(\beta-1)/2}}\right]^{1/\beta}, \sigma_{v} = 1 \tag{28}$$

The distribution of β parameters in determining the main part. The parameter β controls the probability of the shape so that you can get the probability distribution of different shapes, especially in the tail region according to the parameter β. Thus, the smaller the beta parameter, the longer the distribution jumps because there are long tails.

2.6 Opposition-based learning

The opposition-based learning can be regarded as a well-regarded mathematical concept among the community of computational intelligence. The OBL can attain the opposite locations for candidate solutions for a given task. The new location can provide a new chance to become aware of a neighboring point to the best position [41].

The core conception of OBL optimization is, for disclosing an improved solution, simultaneously calculating and evaluating a candidate solution and related matching opposite solution, choosing the best solution as the next-generation individual. For a candidate solutionX_i, the related matching opposite solution$X_i^{'}$ can be calculated according to the following formula:

$$X_i^{'} = a + b - X_i, X_i \in [a, b] \tag{29}$$

Where a and b are the lower bound and upper bound of the search space, respectively. Optimization based optimization: Let X_i be a point in the d-dimensional space, and suppose $f(X_i)$ is a fitness function to evaluate the fitness of candidates. By definition of vertices, $X_i^{'}$ is the opposite of X_i. If $f\left(X_i^{'}\right)$ is better than $f(X_i)$, then update X_i with $X_i^{'}$; Otherwise, keep the current point X_i. Hence, the current point and its relative points are calculated at the same time to be consistent with more appropriate points.

3 Proposed method

3.1 Improved thermal exchange optimization (ITEO)

In this subsection, we describe in detail strategic approaches to improving TEO algorithms. LF and OBL can improve the optimal position moving step length, so

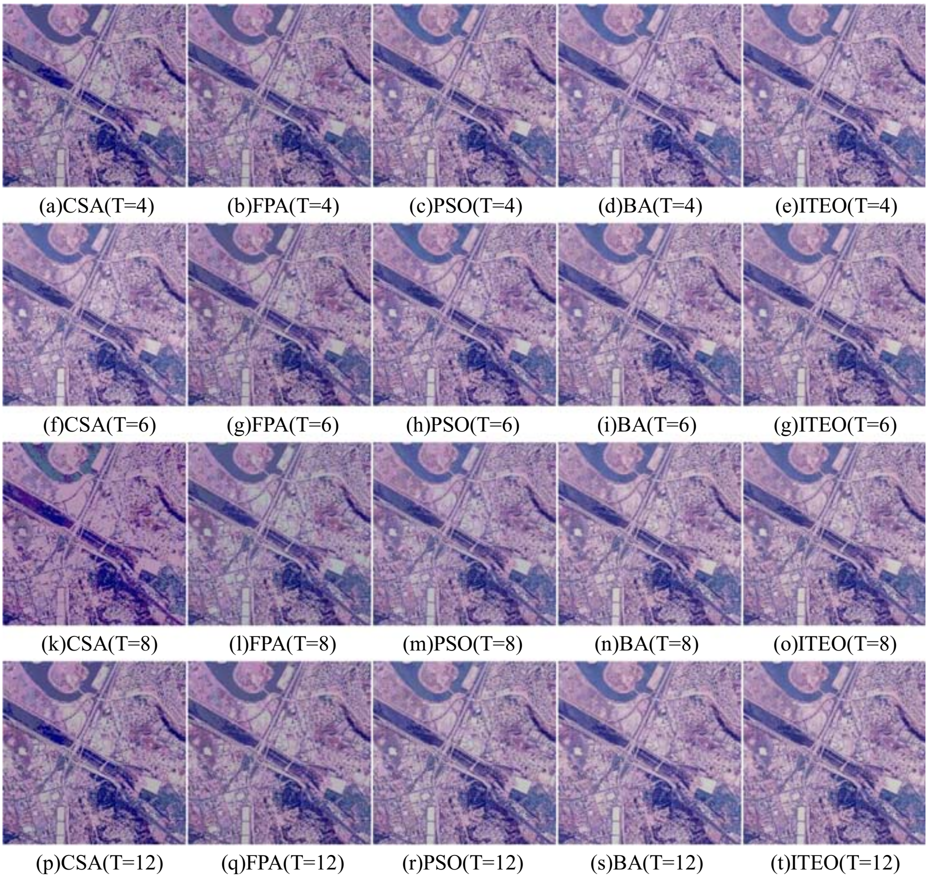


Fig. 5 The segmentation results of Satellite image1

that it can get closer to the food source more quickly. The strategy method can make a suitable balance between the exploration and exploitation. The basic formula of the strategy is shown in section 2. The improved ITEO algorithm will be described by the following formula.

We choose Levy flight with strong randomness to improve the best individual position, increase its jump step size. Levy's flight strategy can speed up the transfer of heat between objects in TEO algorithm, so as to quickly move to the optimal value of the function, which is formulated as follows:

$$T_i^{Levy} = T_i^{env} + \left(T_i^{old} - T_i^{env}\right)\exp(-\beta t) \times Levy(\beta) \tag{30}$$

After updating the positions of all search agents in the population, the opposition-based learning strategy is used to generate the oppositional population corresponding to the current population. This mechanism helps to search for more efficient space and improves the overall exploration capability of the algorithm. Equation 25 can be generated by the following formula:

$$T_i^{op} = T_{\max} + T_{\min} - T_{best} + r(T_{best} - T_i) \tag{31}$$

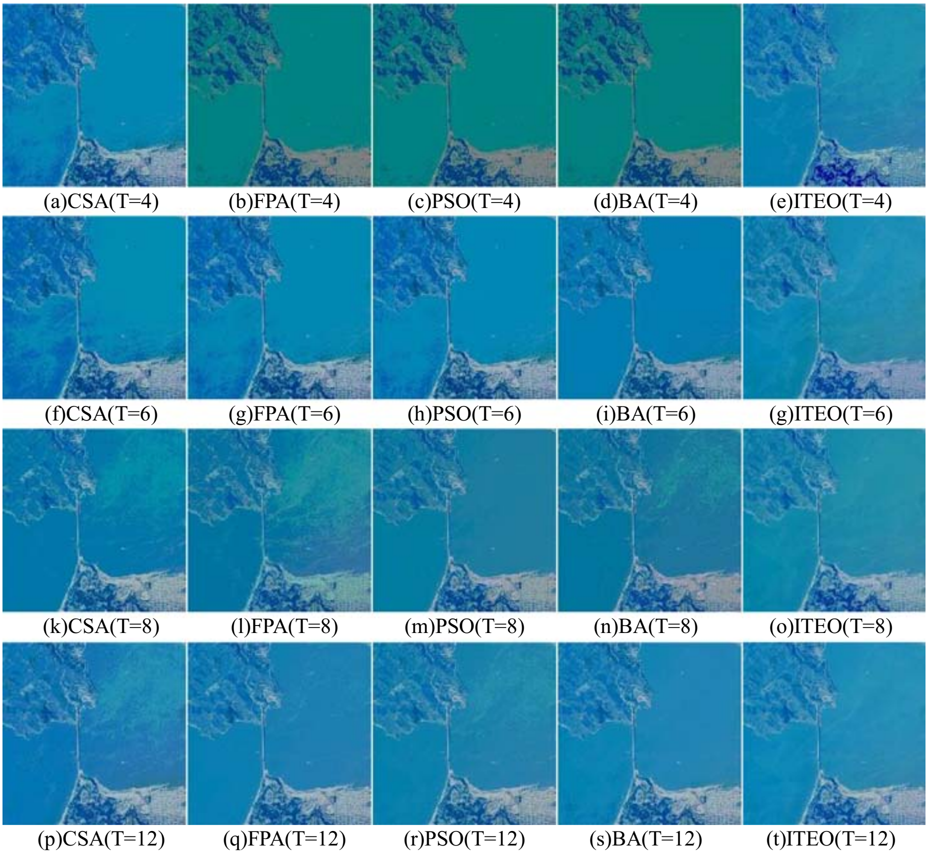


Fig. 6 The segmentation results of Satellite image2

Where, T_i^{op} is the position of the ith opposite temperature inside the search domain. $T_{\max}$and $T_{\min}$ are the lower and upper bounds of the ith variable. T_{best} is the position of the best temperature, r is a random vector with elements inside (0,1). And T_i is the position vector of the ith temperature in population. Then, the best temperature is also updated based on the fitness of the opposite locations. The improved algorithm can better improve the global search ability and convergence performance of TEO algorithm, so as to make the temperature change faster and find the optimal value better.

3.2 Proposed GLCM-ITEO method

In this section, the multi-segmentation method based on ITEO is described in detail. The computational complexity of the proposed method ITEO-GLCM depends on the number of each combination (L), the number of threshold (K), the number of generations (g), the population number (n) and the parameters dimensions (d). Therefore, the overall computational complexity is O(GLCM, ITEO) = g*(O(Updating the position of all search agents) + O(Evaluate the fitness of all agents) + O(Calculate the oppositional position of all search agents and evaluate its fitness) + O(Sort search agents in population and oppositional population)). As we all know, GLCM's

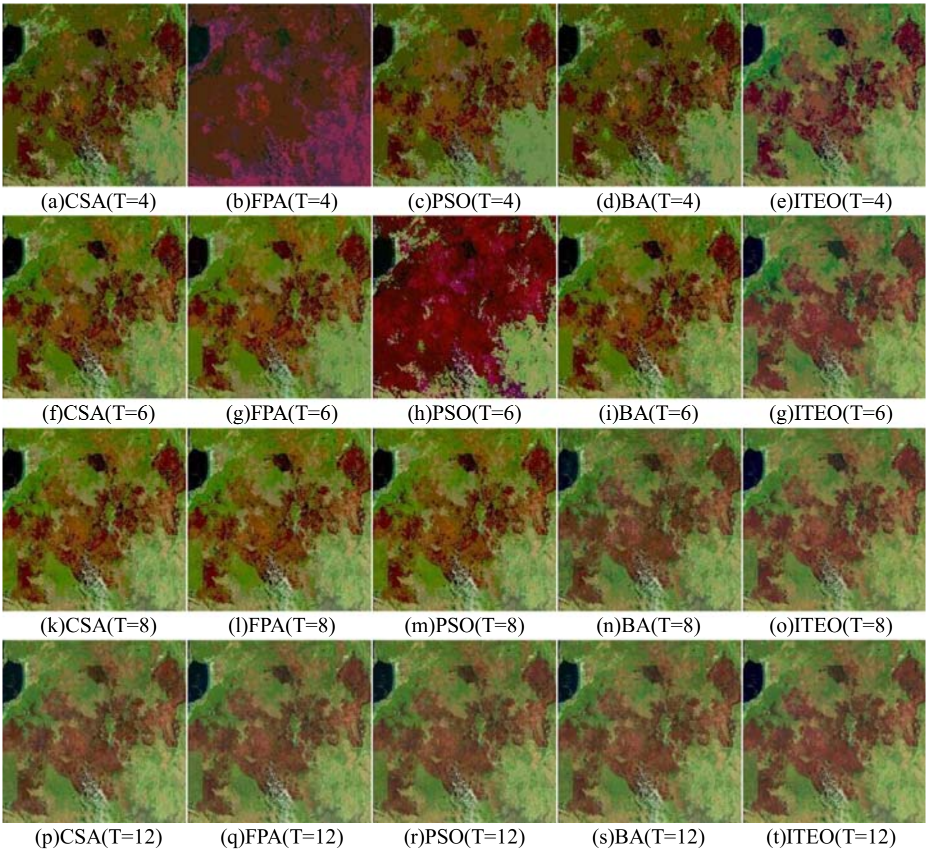


Fig. 7 The segmentation results of Satellite image3

computational complexity of L combination is $O(L^K)$. The computational complexity of updating the position of all search agents is O(n*d). Evaluating the fitness of all agents is $O(n * L^K)$. Calculating the oppositional position of all search agents and evaluate its fitness is $O(n * L^K)$. Sorting search agents in population and oppositional population is O(2n*log2n). So, the final computational complexity of the proposed method is as follow:

$$\begin{aligned} O(GLCM, ITEO) &\approx O\left(g^*\left(n^*d + n^*L^K + n^*L^K + 2n^*\log 2n\right)\right) \\ &= O\left(n^*g^*\left(d + 2\left(L^K + \log 2n\right)\right)\right) \end{aligned} \tag{32}$$

As can be seen from Eq. 32, as the number of thresholds increases, the computational complexity increases, and the segmentation accuracy of the obtained results will decline to some extent. Therefore, ITEO optimization algorithm is used to optimize multi-threshold GLCM, and DCE is used as the fitness function to find the minimum value of this function. At this point, the optimal value obtained is multiple thresholds of the image, and the input image is segmented into multiple regions by multiple thresholds to obtain the segmentation result graph. The flow chart of the ITEO-GLCM can be seen from Fig. 3.

The pseudo code of the GLCM-ITEO is given below.

Algorithm 2 GLCM-ITEO

Begin

Initialize the temperature $T_i(i=1,2,\ldots,n)$;

Initialize *cmax,cmin,*and *maximum number of iterations;*

Cost=the best search agent by Eq.10;

While (l<Max number of iterations)

Update t and β

for each search agent

Update the position of the current search agent by the Eq.30;

Perform Levy flight strategy according to Eq.26;

Bring the current search agent back if it goes outside the boundaries by Eq.31;

end for

Evaluate the fitness of all agents by image thresholding with agent parameters;

Update Cost if there is a better solution;

l= l+1

end while

Return Cost as the optimal parameter for image thresholding;

End

4 GLCM image segmentation experiment

In this section, ITEO algorithm is applied to optimize the DCE function of GLCM algorithm. In order to better verify the image segmentation ability of GLCM-ITEO algorithm, it is compared with the optimized GLCM algorithm of CSA, PSO, FPA and BA. The color image has three color channels. In this paper, the images of the three channels are segmented, and then the three resulting images are fused to obtain the final segmentation result graph. Firstly, the segmentation effect and precision of GLCM-ITEO algorithm are analyzed when the threshold value is increased. Then the segmentation ability, statistical analysis and stability analysis of the proposed ITEO algorithm and other optimization algorithms in GLCM image segmentation are analyzed. Finally, the Berkeley image library is tested and analyzed. All parameters of the comparison optimization algorithm are shown in Table 1.

The test images in this paper are as follows Fig. 4. The test images included color natural images and satellite images. Natural color test images (Kodim images) are accessed from http://r0k.us/graphics/kodak/. The satellite images such as Satellite image1 and Satellite image2 has been obtained from the aerial dataset available on http://sipi.usc.edu/database/database.php?volume=aerials. Satellite image3 and Satellite image4 has been obtained from https://landsat.visibleearth.nasa.gov/. Color image segmentation requires a higher threshold level, so it is more complex to use optimization technology to solve the problem. Therefore, the optimization algorithm has the characteristics of randomness. So, all image segmentation experiments were run separately for 30 times. And the threshold levels of 4, 6, 8 and 12 are selected to find the threshold points corresponding to each color channel in the image. The evaluation of image segmentation result graph is very important, so this paper

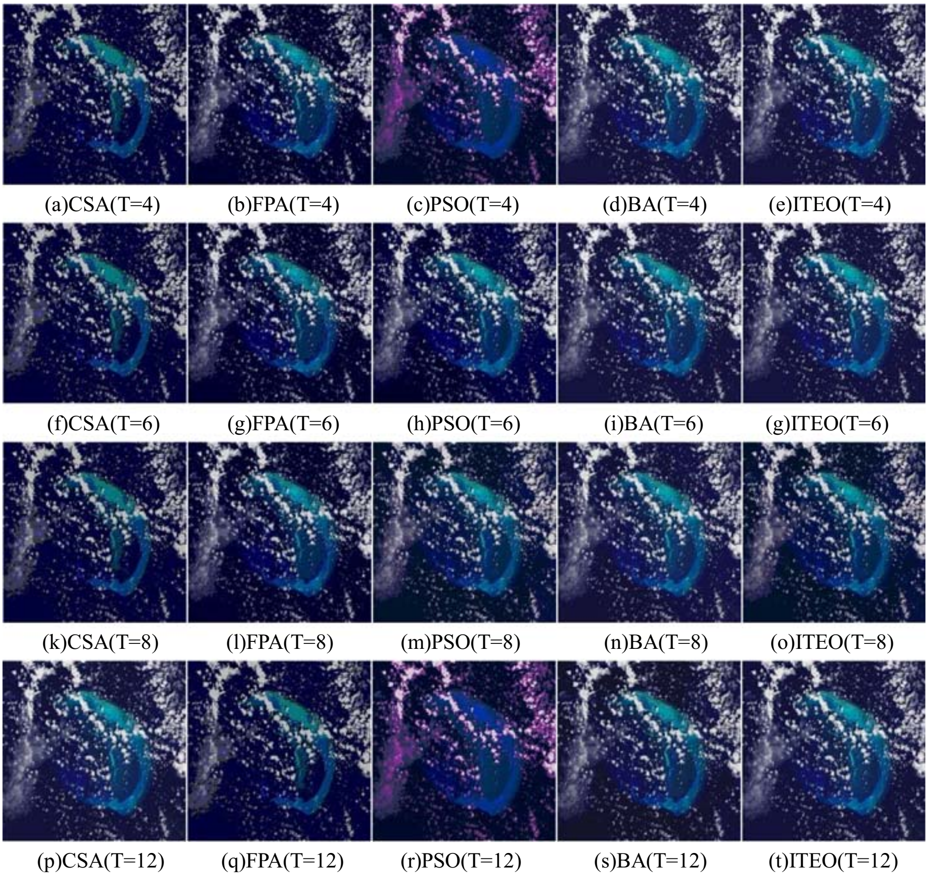


Fig. 8 The segmentation results of Satellite image4

selected Average Precision (AP) [11] and Intersection over Union (IoU) [17]as the evaluation index of test image.

4.1 Experiment 1: Levy's flight parameter β selection

According to the section 3, Levy's flight strategy has a strong jumping ability, and its parameters can affect its jumping results. Therefore, ITEO algorithm is tested. Through experiments on 8 images, AP and IoU values of different β results are obtained in Tables 2 and 3. From the tables, it can be seen that the addition of levy flight effectively improves the segmentation accuracy of the algorithm and increases the optimization ability of TEO algorithm. At the same time, from different β results, it can be clearly seen that they have different influences on the results. Obviously from the table, when the parameter $\beta = 0.5$, the step size is small, and the jumping ability is not obvious, it is easy to fall into the local optimal. When parameter $\beta = 2$, the step size is too large and the jumping ability is too strong, which easily

Fig. 9 The segmentation results of Kodim image1

influences the optimization ability of the algorithm beyond the boundary. When parameter β = 1.5, ITEO gets the best result. So, in subsequent experiments, the parameter β = 1.5.

4.2 Experiment 2: Multilevel thresholding results on GLCM-ITEO

In this experiment, the results obtained by proposed GLCM based ITEO algorithm is analyzed at number of threshold values (T = 4, 6, 8, and 12) for the test images. Satellite images are difficult to be segmented because of their multimodal characteristics. Therefore, an algorithm based on spatial correlation is proposed to solve these problems. Table 4 indicates the AP and IoU values of the segmented results. Higher values of AP and IoU signify better and accurate segmentation. When the number of threshold values T = 4, the AP value and IoU value of each algorithm are lower. With the increase of the number of threshold values, the IOU and AP values also increase, indicating that the increase of the number of threshold values can increase the segmentation precision of the image and make the segmentation result more similar to the original image. Tables 5 and 6 show the optimal thresholds obtained by the proposed technique for satellite images and natural color images, respectively.

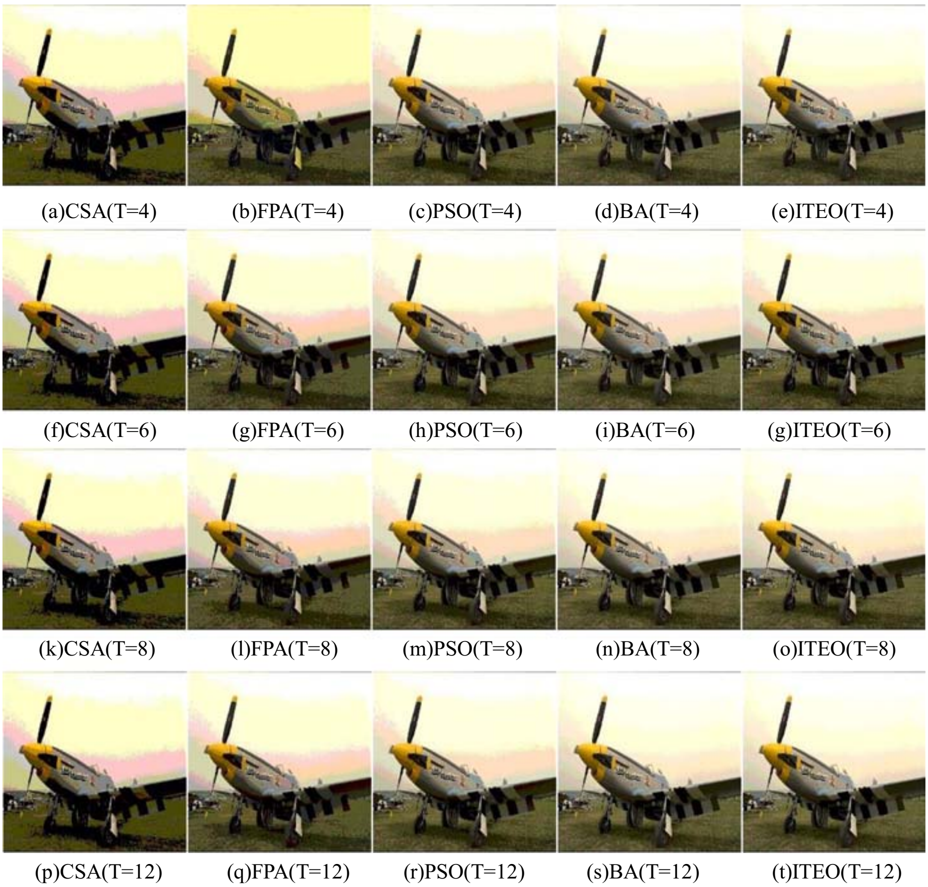


Fig. 10 The segmentation results of Kodim image2

For visual qualitative analysis, the performance of this method at different segmentation levels is shown in Figs. 5, 6, 7, 8, 9, 10, 11 and 12. As can be seen from the satellite images, the ITEO algorithm can also achieve satisfactory segmentation effect with good edge preservation. It can be seen from the natural images that the ITEO algorithm can avoid the phenomenon of under-segmentation and the segmentation of natural images is relatively complete.

4.3 Experiment 3: Comparison with CSA, FPA, PSO, and BA algorithm based multilevel segmentation techniques

In this experiment, to show the merits of proposed GLCM-ITEO technique, the results are compared with CSA, FPA, PSO and BA using same objective function (GLCM). From Table 4, it can be observed that for all the test images, ITEO is better and more reliable than CSA, FPA, PSO, and BA, because of its precise search capability, at a high threshold level (T). Performance of CSA and BA has closely followed ITEO. The solution update strategy for FPA and PSO may have led to poor results. The good results based on the ITEO algorithm are shown in Table 4, and the GLCM-ITEO algorithm performs best in color images such as satellite images. The comprehensive performance ranking of the comparison algorithm is as

Fig. 11 The segmentation results of Kodim image3

follows: ITEO>CSA > BA>FPA > PSO. Tables 5 and 6 shows the optimal threshold of the algorithm for satellite image and natural color image respectively. Therefore, ITEO has the best performance, so it determines the best threshold to produce accurate and high-quality segmentation images.

From Fig. 5, 6, 7, 8, 9, 10, 11 and 12, the visual results show that this method achieves a good segmentation effect by accurately identifying the complex target and background in each level of satellite image segmentation. The image segmentation effect in Figs. 5b, d and 6b–d is poor, and the contour segmentation in satellite images is not clear. As the number of thresholds increases, the image segmentation quality can be enhanced from Figs. 5 and 6. The ITEO algorithm in this paper has the best segmentation effect. It can be seen from Figs. 10, 11 and 12, ITEO algorithm for natural color image segmentation effect is best, CSA and BA algorithm is essentially the same as a result, PSO algorithm segmentation results figure effect is the worst, under segmentation phenomenon exists, the target area segmentation effect is not obvious, and the existence chromatism, the best threshold segmentation results are local optimal phenomenon.

(a)CSA(T=4) (b)FPA(T=4) (c)PSO(T=4) (d)BA(T=4) (e)ITEO(T=4)
(f)CSA(T=6) (g)FPA(T=6) (h)PSO(T=6) (i)BA(T=6) (g)ITEO(T=6)
(k)CSA(T=8) (l)FPA(T=8) (m)PSO(T=8) (n)BA(T=8) (o)ITEO(T=8)
(p)CSA(T=12) (q)FPA(T=12) (r)PSO(T=12) (s)BA(T=12) (t)ITEO(T=12)

Fig. 12 The segmentation results of Kodim image4

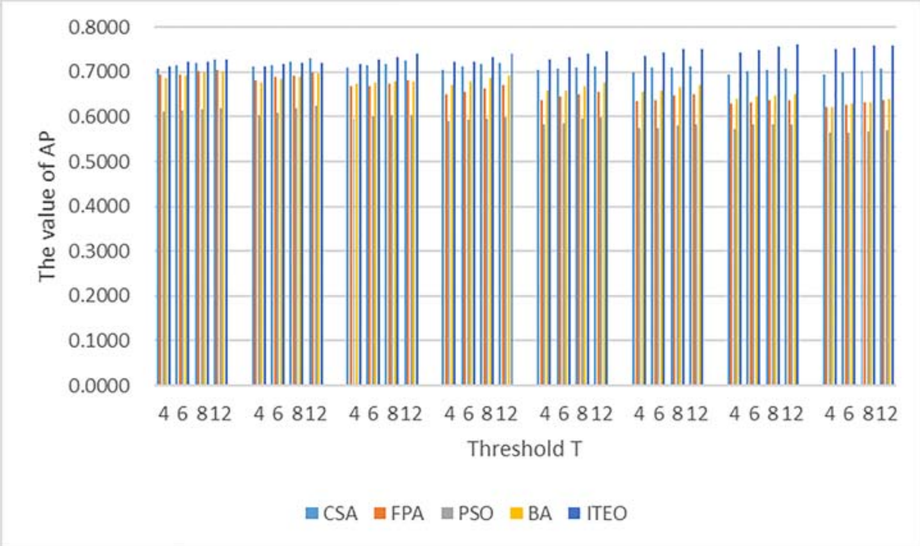


Fig. 13 The histogram of the AP

In order to observe the segmentation performance of each algorithm more intuitively, the histogram of AP and IOU values of algorithm results as shown in Figs. 13 and 14. It can be clearly seen from the figures that ITEO algorithm has a good segmentation ability, which is significantly better than other comparison algorithms. Although the segmentation accuracy of the algorithm is important, the running time of

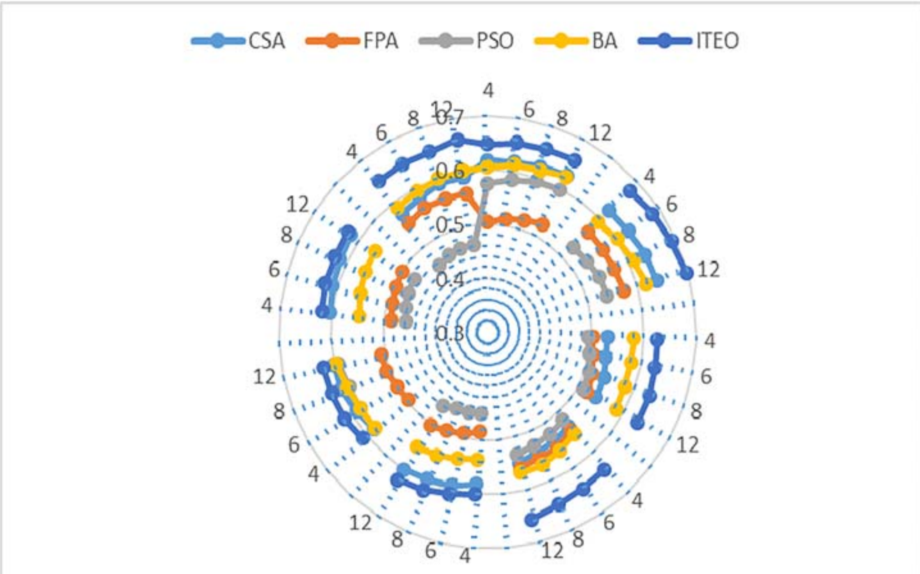


Fig. 14 The histogram of the IoU

the algorithm also affects the segmentation effect. The CPU time of each algorithm is shown in Fig. 15. In order to better observe the performance of each algorithm, the result in the figure is the CPU time used when the threshold number T = 12. As can be seen from Fig. 15, ITEO algorithm has the shortest CPU time, BA algorithm and CSA algorithm have basically the same running time, and FPA algorithm and PSO algorithm have the slowest running time. So, the ITEO algorithm not only has a strong segmentation ability, but also its CPU time is less.

4.4 Stability analysis

Based on the natural optimization algorithm, the results of each run are not the same. Therefore, in order to analyze the stability of the proposed algorithm based on GLCM-ITEO, we use the value of standard deviation (STD). The STD can be intuitive to the operation stability of the algorithm, and the lower the value of the algorithm, the stronger the robustness of the algorithm. Table 7 shows the STD values of each algorithm after 30 runs. It can be seen from the table that the stability of ITEO algorithm is the strongest, especially when dealing with the segmentation of satellite images, its stability is obviously better than other comparison algorithms, indicating that GLCM-ITEO algorithm has a good segmentation ability, and can find the optimal threshold of image better, more accurately and more stably.

4.5 Statistical analysis

We statistically analyze the experimental results to better observe the differences between algorithms. We use Wilcoxon rank sum test [42], a nonparametric statistical test that checks whether one of two independent samples is larger than the other. We calculate the p value of

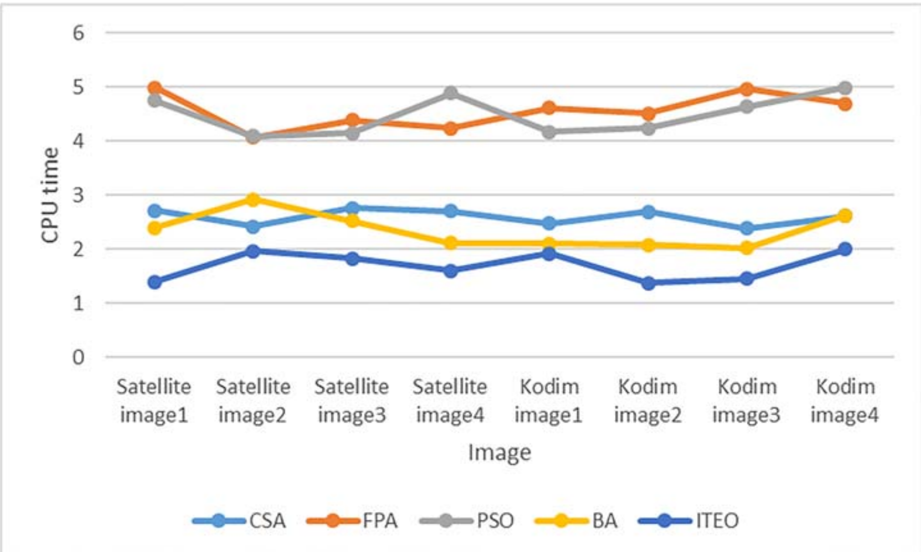


Fig. 15 The histogram of the CPU time

Table 7 Comparison of standard deviation (STD) of IoU computed by CSA, FPA, PSO, BA and ITEO using GLCM as an objective function

Test Images	T	CSA	FPA	PSO	BA	ITEO
Satellite image1	4	5.92E-08	3.99E-09	3.92E-08	6.44E-10	**5.80E-16**
	6	4.26E-09	5.25E-08	1.34E-08	4.96E-09	**2.70E-12**
	8	9.28E-09	4.30E-08	4.06E-08	1.33E-11	**4.17E-13**
	12	3.51E-08	9.42E-08	4.12E-08	7.56E-08	**3.33E-15**
Satellite image2	4	6.01E-08	1.16E-08	7.29E-08	1.76E-08	**1.54E-15**
	6	2.67E-08	2.99E-08	3.92E-08	1.03E-08	**5.94E-16**
	8	2.82E-08	2.75E-08	4.20E-10	1.82E-08	**1.42E-15**
	12	4.27E-08	1.85E-10	1.74E-08	3.15E-06	**2.64E-16**
Satellite image3	4	6.05E-09	1.03E-08	1.87E-08	4.26E-08	**9.65E-11**
	6	1.34E-09	3.95E-09	3.48E-02	5.69E-08	**4.24E-15**
	8	1.49E-08	1.63E-08	2.94E-02	7.76E-09	**4.76E-16**
	12	8.52E-08	2.62E-08	2.72E-02	2.78E-09	**4.73E-11**
Satellite image4	4	2.04E-09	2.63E-08	4.77E-02	2.36E-09	**2.91E-12**
	6	2.40E-08	7.17E-08	3.36E-08	5.81E-08	**1.28E-11**
	8	3.78E-09	1.29E-08	5.90E-08	3.32E-05	**3.39E-16**
	12	4.56E-08	1.84E-08	3.82E-04	7.39E-08	**4.48E-14**
Kodim image1	4	3.05E-08	4.81E-08	1.39E-08	7.72E-08	**2.21E-15**
	6	4.63E-08	2.02E-08	4.85E-08	8.86E-08	**3.31E-12**
	8	5.65E-08	6.23E-08	4.91E-06	2.11E-08	**1.36E-11**
	12	6.29E-08	1.42E-08	2.28E-08	5.91E-09	**1.27E-16**
Kodim image2	4	1.38E-08	6.91E-08	6.09E-08	6.76E-08	**3.37E-14**
	6	1.38E-05	5.01E-09	8.55E-09	1.08E-08	**1.10E-11**
	8	7.87E-10	4.17E-10	1.70E-02	8.90E-08	**4.74E-13**
	12	1.05E-09	4.06E-08	3.50E-08	7.40E-09	**4.40E-12**
Kodim image3	4	1.22E-08	2.15E-05	3.69E-10	1.49E-09	**3.07E-10**
	6	5.56E-08	2.00E-02	5.72E-03	7.08E-05	**1.52E-14**
	8	7.37E-09	1.19E-10	5.87E-09	1.52E-08	**1.88E-13**
	12	5.49E-09	9.52E-10	5.25E-08	4.10E-08	**1.51E-12**
Kodim image4	4	1.00E-07	5.25E-05	3.14E-09	1.15E-08	**1.15E-11**
	6	9.23E-09	5.87E-08	1.01E-08	3.10E-08	**8.50E-11**
	8	7.75E-08	1.05E-08	2.49E-08	9.28E-05	**3.47E-13**
	12	7.48E-08	4.35E-08	1.19E-09	6.37E-08	**4.16E-13**

IoU of ITEO algorithm and CSA, FPA, PSO and BA algorithm. The experimental statistical results are shown in Table 8.

If the p value of two algorithms is greater than 0.05, there is no significant difference between the two algorithms. On the other hand, a p value less than 0.05 means that there is a significant difference between the two algorithms at the significance level of 5%. It can be seen from the Table 8 that 28 out of 32 results of ITEO algorithm are better than CSA algorithm, 32 out of 32 results are better than FPA algorithm, 32 out of 32 results are better than PSO algorithm, and 30 out of 32 results are better than BA algorithm. Therefore, the GLCM-ITEO algorithm is obviously better than the comparison algorithm in the statistical sense.

4.6 Comparison of different algorithms on Berkley segmentation data set (BSDS300)

Each multilevel image thresholding method has also been evaluated using a well-known benchmark-the Berkley segmentation data set (BSDS300) with 300 distinct images. The 300

Table 8 The calculated p values from the Wilcoxon test for the GLCM-ITEO versus other optimizers

Test Images	T	CSA	FPA	PSO	BA
Satellite image1	4	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P > 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Satellite image2	4	$P > 0.05$	$P > 0.05$	$P < 0.05$	$P < 0.05$
	6	$P > 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Satellite image3	4	$P > 0.05$	$P < 0.05$	$P < 0.05$	$P > 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Satellite image4	4	$P > 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Kodim image1	4	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Kodim image2	4	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Kodim image3	4	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
Kodim image4	4	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	6	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	8	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$
	12	$P < 0.05$	$P < 0.05$	$P < 0.05$	$P < 0.05$

images from the Berkeley segmentation data set (BSDS 300) available at https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images.html. This paper uses an extensive comparative study on Berkeley database by using performance metrics like Probability Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE) [2, 43, 44]. Table 9 shows the average results of PRI, BDE, GCE and VoI of ground truth results of the 300 images of BSDS300 data set.

The results displayed in Table 9, that the proposed technique outperforms all other compared multilevel thresholding algorithms. The GLCM-ITEO technique has obtained results close to the ground truth images. Higher values of PRI indicate better segmentation performance. While lower values of BDE, GCE, and VoI show better segmentation. It can be seen from the table that the numerical value of GLCM-ITEO algorithm is the best, indicating that its segmentation result is the closest to groundtruth and the segmentation effect is the best. And the PSO and FPA algorithm segmentation effect is the most check. So, GLCM-ITEO algorithm can effectively solve the problem of image segmentation.

Table 9 The comparison results for the GLCM-ITEO versus other optimizers

Algorithm	T	BDE	PRI	GCE	VOI
Ground truth		5.5862	0.9658	0.0906	1.0121
GLCM-CSA	4	9.9772	0.6168	0.3325	4.5721
	6	9.1575	0.6367	0.3050	4.5941
	8	9.2713	0.6503	0.3868	4.5919
	12	9.1088	0.6286	0.3268	4.8687
GLCM-FPA	4	10.6636	0.3164	0.4323	6.6578
	6	10.1924	0.3564	0.4861	6.7498
	8	10.0210	0.3700	0.4205	6.6354
	12	10.6942	0.3255	0.4107	6.5959
GLCM-PSO	4	11.8489	0.3966	0.5169	7.7391
	6	11.1235	0.3087	0.5329	7.4703
	8	11.6988	0.3592	0.5670	7.8015
	12	11.0201	0.3818	0.5249	7.7039
GLCM-BA	4	9.6466	0.5276	0.4044	4.5414
	6	9.5655	0.5438	0.4846	4.1608
	8	9.9093	0.5948	0.4078	4.7028
	12	9.3861	0.5977	0.4486	4.5979
GLCM-ITEO	4	**8.3161**	**0.7774**	**0.2586**	**3.2826**
	6	**8.0681**	**0.7077**	**0.2418**	**3.6486**
	8	**8.6627**	**0.7632**	**0.2357**	**3.4191**
	12	**8.7981**	**0.7908**	**0.2936**	**3.9171**
Otsu-CSA	4	10.3119	0.5611	0.3406	5.8017
	6	10.4464	0.5126	0.3909	5.3104
	8	10.4641	0.5261	0.3292	5.6079
	12	10.0146	0.5417	0.3005	5.1237
Otsu -FPA	4	11.6080	0.3723	0.3867	5.2668
	6	11.1117	0.3852	0.3301	5.0957
	8	11.4411	0.3259	0.3177	5.6213
	12	11.9995	0.3148	0.3014	5.4527
Otsu -PSO	4	11.5348	0.4166	0.4189	6.2575
	6	11.2996	0.4539	0.4393	6.8886
	8	11.6907	0.4559	0.4046	6.2144
	12	11.6169	0.4450	0.4776	6.0834
Otsu -BA	4	9.2444	0.5232	0.3284	5.9099
	6	9.0448	0.5823	0.3262	5.5672
	8	9.1535	0.5084	0.3223	5.3078
	12	9.7935	0.5402	0.3245	5.4523
Otsu -ITEO	4	9.2361	0.6958	0.3605	4.0933
	6	9.2402	0.6684	0.3160	4.6615
	8	9.7611	0.6024	0.3680	4.0908
	12	9.0914	0.6802	0.3469	4.5156

5 Conclusions

In this paper, the ITEO algorithm is used to optimize the multi-threshold GLCM algorithm to obtain the optimal multi-threshold image. We use LF and OBL strategies to improve TEO algorithm, increase the random step size of the algorithm, and improve the optimization ability of the algorithm. In this paper, the algorithm is compared with other optimization algorithms to jointly optimize GLCM algorithm for color natural image and satellite image segmentation experiments. From AP and IoU values, it can be seen that GLCM-ITEO algorithm has the best segmentation accuracy. Finally, we compare the GLCM-ITEO algorithm with the multi-threshold Otsu algorithm, and conduct segmentation experiments on 300 images in the

Berkeley image library. It can be seen from the PRI, VoI, GCE and BDE index that both the ability of ITEO to optimize GLCM and Otsu algorithm is better than other comparative optimization algorithms, and the segmentation effect of GLCM-ITEO algorithm is better than the segmentation effect of multi-threshold Otsu algorithm. Therefore, the GLCM-ITEO algorithm proposed in this paper has better image segmentation accuracy and better stability. In the future, we will continue to study multi-threshold methods and different optimization algorithms, so as to improve the image segmentation accuracy.

References

1. Ahmed AE, Mohamed AE, Essam HH (2018) Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 112:156–172
2. Akram F, Garcia MA, Puig D (2017) Active contours driven by local and global fitted image models for image segmentation robust to intensity inhomogeneity. PLoS One 12(4):e0174813
3. Askarzadeh, Alireza (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12
4. Becker AS, Wagner MW, Wurnig MC et al (2017) Diffusion-weighted imaging of the abdomen: impact of b-values on texture analysis features. NMR Biomed 30(1):e3669
5. Biyanto TR et al (2017) Killer whale algorithm: an algorithm inspired by the life of killer whale. Proc Comput Sci 124:151–157
6. Bouchekara HREH, Chaib AE, Abido MA et al (2016) Optimal power flow using an improved colliding bodies optimization algorithm. Appl Soft Comput 42(C):119–131
7. Chen Y, Zhang Y, Yang J (2016) Curve-like structure extraction using minimal path propagation with backtracing. IEEE Trans Image Process 25(2):988–1003
8. Cheng X, Shuai CM, Wang J et al (2018) Building a sustainable development model for China's poverty-stricken reservoir regions based on system dynamics. J Clean Prod 176:535–554
9. Dong S, Li H, Wang J et al (2017) Improved flexible Li-ion hybrid capacitors: techniques for superior stability. Nano Res 10(12):4448–4456
10. Dong W, Kang L, Zhang W (2017) Opposition-based particle swarm optimization with adaptive mutation strategy. Soft Comput 21(17):5081–5090
11. Fadaei S, Amirfattahi R, Ahmadzadeh MR (2017) New content-based image retrieval system based on optimised integration of DCD, wavelet and curvelet features. IET Image Process 11(2):89–98
12. Fan L, Clausi DA, Xu L et al (2018) ST-IRGS: a region-based self-training algorithm applied to hyperspectral image classification and segmentation. IEEE Trans Geosci Remote Sens 56(1):3–16
13. Farag TH, Hassan WA, Ayad HA et al (2017) Extended absolute fuzzy connectedness segmentation algorithm utilizing region and boundary-based information. Arab J Sci Eng 42(8):3573–3583
14. Fister I Jr et al (2014) A novel hybrid self-adaptive bat algorithm. TheScientificWorldJournal 1–2:709738
15. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
16. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99
17. Hazirbas C et al (2016) FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture. In: Asian conference on computer vision (ACCV). Springer, Cham
18. He Y, Chiu WC, Keuper M et al (2017) STD2P: RGBD semantic segmentation using spatio-temporal data-driven pooling// Computer Vision & Pattern Recognition
19. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Appl Soft Comput 60:115–134
20. Hong KS, Khan MJ (2017) Hybrid brain–computer interface techniques for improved classification accuracy and increased number of commands: a review. Front Neurorobot 11:35
21. Jiang Y, Yeh WC, Hao Z et al (2016) A cooperative honey bee mating algorithm and its application in multi-threshold image segmentation. Inf Sci 369(1):171–183
22. Kang Y, Lee GY, Lee JW, Lee E, Kim B, Kim SJ, Ahn JM, Kang HS (2017) Texture analysis of torn rotator cuff on preoperative magnetic resonance arthrography as a predictor of postoperative tendon status. Korean J Radiol 18(4):691–698
23. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84

24. Kennedy J, Eberhart R (2002) Particle swarm optimization. In: Proceedings of ICNN'95 – international conference on Neural Networks. IEEE
25. Kwong STW, Gao H, Pun CM et al (2018) An improved artificial bee colony algorithm with its application to metallographic image segmentation. IEEE Trans Ind Informatics PP(99):1–1
26. Leszczyński B, Gancarczyk A, Wróbel A et al (2016) Global and local thresholding methods applied to X-ray microtomographic analysis of metallic foams. J Nondestruct Eval 35(2):35
27. Li M, Liao JJ (2012) Texture image segmentation based on GLCM. Appl Mech Mater 220–223:1398–1401
28. Li Y, Bai X, Jiao L et al (2017) Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Appl Soft Comput 56(C):345–356
29. Li H et al (2017) A novel unsupervised Levy flight particle swarm optimization (ULPSO) method for multispectral remote-sensing image classification. Int J Remote Sens 38(23):6970–6992
30. Lv T, Yang G, Zhang Y, Yang J, Chen Y, Shu H, Luo L (2019) Vessel segmentation using centerline constrained level set method. Multimed Tools Appl 78:17051–17075. https://doi.org/10.1007/s11042-018-7087-x
31. Mala C, Sridevi M (2016) Multilevel threshold selection for image segmentation using soft computing techniques. Soft Comput 20(5):1793–1810
32. Malegori C, Franzetti L, Guidetti R et al (2016) GLCM, an image analysis technique for early detection of biofilm. J Food Eng 185:48–55
33. Marinaki M, Marinakis Y (2016) A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands. Expert Syst Appl 46(C):145–163
34. Mesa A, Castromayor K, Garillos-Manliguez C et al (2017) Cuckoo search via Levy flights applied to uncapacitated facility location problem. J Ind Eng Int 14(3):585–592
35. Mirjalili S (2015) The Ant Lion Optimizer. Adv Eng Softw 83(C):80–98
36. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human mental search: a new population-based metaheuristic optimization algorithm. Appl Intell 47(3):850–887
37. Niu S, Qiang C, Sisternes LD et al (2017) Robust noise region-based active contour model via local similarity factor for image segmentation. Pattern Recogn 61:104–119
38. Oghaz MM, Maarof MA, Rohani MF et al (2017) An optimized skin texture model using gray-level co-occurrence matrix. Neural Comput Applic 6:1–19
39. Oliva D, Hinojosa S, Cuevas E, Pajares G, Avalos O, Gálvez J (2017) Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. Expert Syst Appl 79:164–180. https://doi.org/10.1016/j.eswa.2017.02.042
40. Qayyum R, Kamal K, Zafar T et al (2016) Wood defects classification using GLCM based features and PSO trained neural network. In: International conference on Automation & Computing. IEEE
41. Raj S, Bhattacharyya B (2018) Reactive power planning by opposition-based grey wolf optimization method. Int Trans Electr Energy Syst 3:e2551
42. Rosner B, Glynn RJ, Ting LM (2015) Incorporation of clustering effects for the Wilcoxon rank sum test: a large-sample approach. Biometrics 59(4):1089–1098
43. Sarkar S, Das S (2013) Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach. IEEE Trans Image Process 22(12):4788–4797
44. Sarkar S, Das S, Chaudhuri SS (2015) A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution. Pattern Recogn Lett 54:27–35
45. Singh VP, Prakash T, Singhrathore N et al (2016) Multilevel thresholding with membrane computing inspired TLBO. Int J Artif Intell Tool 25(06):1650030
46. Storn R, Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
47. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation, and international conference on intelligent agents, web technologies and internet commerce
48. Vallaeys V et al (2017) A Lévy-flight diffusion model to predict transgenic pollen dispersal. J R Soc Interface 14(126):20160889
49. Xuan TP, Siarry P, Oulhadj H (2018) Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation. Appl Soft Comput 65:230–242
50. Xue J, He X, Yang X et al (2017) Multi-threshold image segmentation method based on flower pollination algorithm. Commun Comput Inf Sci 791:39–51
51. Xutang Z et al (2016) An effective approach of teeth segmentation within the 3D cone beam computed tomography image based on deformable surface model. Math Probl Eng 2016:1–10
52. Yan B et al (2017) A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters. Comput Phys Commun 219:S001046551730139X

53. Ye ZW, Wang MW, Liu W et al (2015) Fuzzy entropy based optimal thresholding using bat algorithm. Appl Soft Comput 31(C):381–395
54. Zhang H, Xie J, Hu Q et al (2018) A hybrid DPSO with Levy flight for scheduling MIMO radar tasks. Appl Soft Comput 71:242–254
55. Zhao M, Zhang X, Shi Z et al (2018) Restoration of motion blurred images based on rich edge region extraction using a gray-level co-occurrence matrix. IEEE Access 6:15532–15540
56. Zheng, Yu-Jun (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55: 1–11
57. Zhou J, Yao X (2017) Multi-objective hybrid artificial bee colony algorithm enhanced with Lévy flight and self-adaption for cloud manufacturing service composition. Appl Intell 47(3):721–742