Engineering Applications of Artificial Intelligence 103 (2021) 104293

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization

Bibekananda Jena [a], Manoj Kumar Naik [b,*], Rutuparna Panda [c], Ajith Abraham [d]

[a] Department of Electronics and Communication Engineering, Anil Neerukonda Institute of Technology & Science, Sangivalasa, Visakhapatnam, Andhra Pradesh 531162, India
[b] Faculty of Engineering and Technology, Siksha O Anusandhan, Bhubaneswar, Odisha 751030, India
[c] Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology, Burla, Odisha 768018, India
[d] Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, WA 98071-2259, USA

ARTICLE INFO

Keywords:
Machine intelligence
Soft computing
Multilevel thresholding
Brain MRI

ABSTRACT

Nevertheless, the accuracy of a multilevel image thresholding technique using 1D or 2D Tsallis entropy is limited. To overcome this, we propose a maximum 3D Tsallis entropy-based multilevel thresholding method. The idea of 3D Tsallis entropy is introduced. Opposed to the 1D/2D Tsallis entropy, the 3D Tsallis entropy-based approach is more robust, it performs well even in the case of the low signal-to-noise-ratio and contrast. Manta Ray Foraging Optimization (MRFO) algorithm is a newly introduced algorithm to solve the optimization problem by imitating the foraging technique of Manta Ray fish in the ocean using a mathematical model. Due to insufficient energy levels of search agents in MRFO, they fail to avoid local minima and fall on it. To make the algorithm more effective for the segmentation application, we introduce a new algorithm coined as attacking Manta Ray foraging optimization (AMRFO). A set of classical benchmark functions together with composite functions (CEC 2014) is used to validate the proposed AMRFO algorithm. Statistical analysis is implicitly carried out using Wilcoxon's signed-rank test and Friedman's mean rank test. Interestingly, the results show that the proposed AMRFO is superior to the state-of-the-art optimization algorithms. Moreover, the proposed method is also compared with 1D/2D Tsallis entropy-based approaches. To experiment, 100 test images from the AANLIB MR Image dataset are considered. Our method outperforms 1D/2D Tsallis entropy-based approaches. The proposed scheme would be useful for the segmentation of multi-spectral color images.

1. Introduction

In the field of image processing and computer vision, image segmentation has its importance. The role of Image segmentation is to decompose the test image into meaningful homogeneous parts or objects depend upon the problem for easier analysis. In most of the applications, the objects of interest in the scene need to be processed for which different types of segmentation techniques are used in the initial stage of processing. Image segmentation has numerous applications in the area of Medical Imaging, Object detection and recognition, Machine vision, etc. (Zaitoun and Aqel, 2015). There are various methods of image segmentation. One of the popular, effective, and convenient methods is the thresholding-based method. In bi-level thresholding, the scene in the image is subdivided into two different regions with the help of a threshold value obtained from the image histogram. Whereas, in multilevel thresholding, two or more thresholds are used to segment the test image into different classes.

One of the challenging tasks of thresholding-based segmentation is to obtain the optimal threshold values. There are two approaches to finding an optimal threshold: parametric approach and Non-parametric approach. Due to the high computational complexity of a parametric-based approach, it is hardly used for multilevel image thresholding (Bohat and Arya, 2019). Therefore, non-parametric methods like histogram-based thresholding became more popular because of their robustness and speed. The threshold calculation in non-parametric thresholding methods generally based on some statistical criteria or entropy measures. One of the well-known statistical criteria-based thresholding is Otsu's Thresholding algorithm (Otsu, 1979) which works on the principle of maximizing between class variance for obtaining the optimal thresholds. Some of the popular entropy-based methods include Tsallis's Entropy (Agrawal et al., 2013), Kapur's Entropy (Kapur et al., 1985), Renyi entropy (Sahoo et al., 1997), Fuzzy entropy (Chao et al., 2016), and Masi Entropy (Masi, 2005; Shubham and Bhandari, 2019) calculate the optimal threshold by maximizing the corresponding

[*] Corresponding author.
E-mail addresses: bibekananda.jena@gmail.com (B. Jena), naik.manoj.kumar@gmail.com (M.K. Naik), r_ppanda@yahoo.co.in (R. Panda), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2021.104293
Received 23 July 2020; Received in revised form 8 April 2021; Accepted 3 May 2021
Available online 12 May 2021

Table 1
Parameter setting of different optimization algorithms.

Algorithm	Parameter(s)	Value
DE	Mutation factor (F) Cross over Rate (Cr)	0.5 0.5
PSO	C_1 *and* C_2 Inertia Weight (ω)	2 0.9
SFO	Coefficient A *and* ϵ	4 and 0.001
EO	Constants a_1 and a_2 Initial Generation value (G_0)	2 and 1 0.5
MRFO	Somersault factor (S)	2
AMRFO	Initial Attacking Power (A) Somersault factor (S)	3 2

entropic information between object pixels and background pixels. One of the major limitations of threshold selection using a 1D histogram is that it does not consider the spatial distribution of the pixels in the image (Nie et al., 2013). Images having identical histograms may have a different spatial distribution which should be treated in the same manner for threshold selection. Therefore, researchers started using the local average of pixel intensity which carries the spatial information of neighborhood pixels in a 2D histogram along with original pixel points for threshold selection. 2D Otsu thresholding (Liu et al., 1991), 2D Tsallis entropy-based thresholding (Sarkar and Das, 2013a), 2D Masi Entropy (Wunnava et al., 2020a), and gray gradient algorithm-based thresholding (Wunnava et al., 2020b) are some well-known algorithms proves that 2D histogram-based methods are better and more effective for noisy images than the 1D histogram. However, 2D histogram-based methods are not always satisfactory because the target object points and the points on the boundary of second and third regions closed to the diagonal line are ignored (Wang et al., 2012).

To overcome this problem, 3D histogram-based thresholding was introduced by including the median values of the neighboring pixels are included in the histogram with average pixel value and original pixel gray level. 3D Otsu function-based thresholding (Bhandari et al., 2019) is one of the recent works on a 3D histogram and claimed that it has better performance than 1D and 2D based histogram method for images in low signal to noise ratio and poor contrast. These advantages of 3D histogram-based threshold selection motivate us to propose a Maximum 3D Tsallis entropy-based multilevel thresholding, as an extension of 2D Tsallis entropy. Although 3D histogram-based techniques are found suitable for segmentation, they are generally consuming more time for processing an image due to high computational complexity.

Evolutionary metaheuristic algorithms are successfully applied to solve the computational time problem of multilevel thresholding algorithms for image segmentation. The most popular meta-heuristic algorithms which are effectively applied in multilevel thresholding are Particle swarm optimization (PSO) (Maitra and Chatterjee, 2008a); Differential evolution (DE) (Horng, 2010; Sarkar and Das, 2013a); Ant colony optimization (ACO) (Zhiwei et al., 2005); Honey bee mating optimization (HBMO); Grasshopper optimization algorithm (GOA) (Saremi et al., 2017); Sailfish optimizer (SFO) (Shadravan et al., 2019), Crow search algorithm (CSA) (Education, 2019); Firefly algorithm (FA) (Horng and Liou, 2011); Artificial bee colony (ABC) (Horng, 2011); Cuttlefish algorithm(CFA) (Bhandari et al., 2019); Cuckoo search algorithm (CS) algorithm (Bhandari et al., 2014); Squirrel search algorithm (SSA) (Jain et al., 2019); Gray wolf optimizer (GWO) (Khairuzzaman and Chaudhury, 2017); Krill herd optimization (KHO) (Baby Resma and Nair, 2018) and Wind driven optimization (WDO) (Bhandari et al., 2014). It has also been observed that some algorithms are modified or hybridized with other algorithms by the researchers to make them more suitable than their original versions. Some of these algorithms are adaptive Harris hawks optimization (AHHO) (Wunnava et al., 2020b); Modified grasshopper optimization algorithm (MGOA) (Liang et al., 2019); Hybrid differential evolution (HDE) (Mlakar et al., 2016); Improved adaptive cuckoo search algorithm (ICAS) (Sun and Wei, 2020); Modified firefly algorithm (MFA) (He and Huang, 2017); Hybrid salp swarm Algorithm (HSSA) (Alwerfali et al., 2019) and Improved bat algorithm (IBA) (Alihodzic and Tuba, 2014).

Manta ray foraging optimization (MRFO) (Zhao et al., 2020) is the recently proposed algorithm based on the foraging behavior of the Manta Ray creature in the ocean. The impressive performance of MRFO against other well-known optimization algorithms draws our attention to it to explore its strength and weakness. According to No Free Lunch (NFL) (Wolpert and Macready, 1997), which stated that no optimization algorithm gives satisfactory results for all kinds of problems and MRFO is not an exception, there is a possibility to improve its global searchability. After a detailed investigation, we introduced additional attacking power to the Manta ray during foraging which helps it to avoid local minima. This leads to the formation of the Attacking manta ray foraging optimization algorithm (AMRFO). Performance of AMRFO found better when tested with a set of 21 classical Benchmark functions (Wunnava et al., 2020b) and six composition test functions from CEC 2014 (Liang et al., 2013) against the well-known state-of-art algorithms such as MRFO (Zhao et al., 2020); EO (Abdel-Basset et al., 2020); HHO (Heidari et al., 2019); SFO (Shadravan et al., 2019); GWO (Khairuzzaman and Chaudhury, 2017); PSO (Maitra and Chatterjee, 2008b) and DE (Sarkar and Das, 2013a). The performance of AMRFO is further explored in this paper to obtain optimal thresholds from AANLIB Brain MR images taken from Harvard medical education (The Whole Brain Atlas) by maximizing Tsallis entropy in a 3D histogram structure.

The important contributions of the proposed work are:

- A 3D Tsallis entropy derived from the 3D histogram is proposed in this paper as an extension of 2D Tsallis entropy for more effective multilevel thresholding in a different situation.
- An attacking manta ray foraging optimization (AMRFO) algorithm is also proposed by including an additional attacking power for the Manta Ray during searching for food, which helps it to reach a global solution without falling in any local minima. The quantitative and qualitative analysis of the test results over 27 different Benchmark functions shows the superiority of the proposed method.
- Maximum 3D Tsallis entropy-based multilevel thresholding using AMRFO is validated using the ANNLIB Brain MRI dataset. The comparison results show its better performance over state-of-the-art methods.

A detailed discussion of the work proposed in this paper is presented in the following sections. The rest part of the paper is organized as follows. The proposed 3D Tsallis entropy-based multilevel thresholding method is presented in Section 2. In Section 3, we discuss the proposed Attacking Manta Ray foraging optimization (AMRFO) algorithm. The performance evolution of the proposed Optimization algorithm against other state-of-the-art methods is presented in Section 4. The framework of 3D Tsallis entropy-based multilevel thresholding using the AMRFO algorithm is presented in Section 5. Results and discussion of Maximum 3D Tsallis entropy-based multilevel thresholding using AMRFO are presented in Section 6. The paper ended with a concluding remark in Section 7.

2. The proposed 3-D Tsallis entropy-based multilevel image thresholding

Tsallis entropy-based multilevel thresholding is one of the efficient and popular methods of image segmentation. In this section, we introduced a new 3D Tsallis entropy, which is the extension of 2D Tsallis Entropy (Sarkar and Das, 2013a) by considering both mean and median value of the neighboring pixels along with pixel intensity in a 3D histogram as shown in Fig. 1(a) into account. This extension of Tsallis

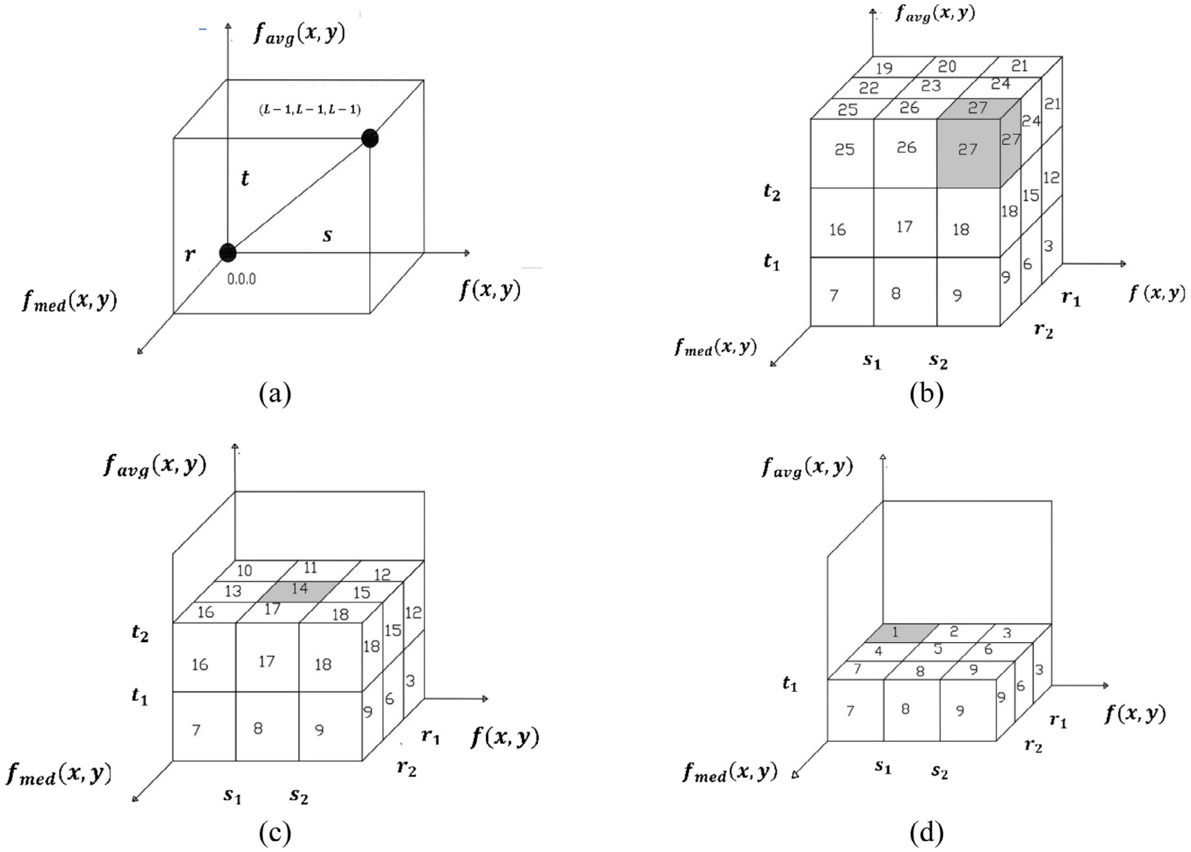

Fig. 1. 3D-histogram structure for 3-level segmentation.

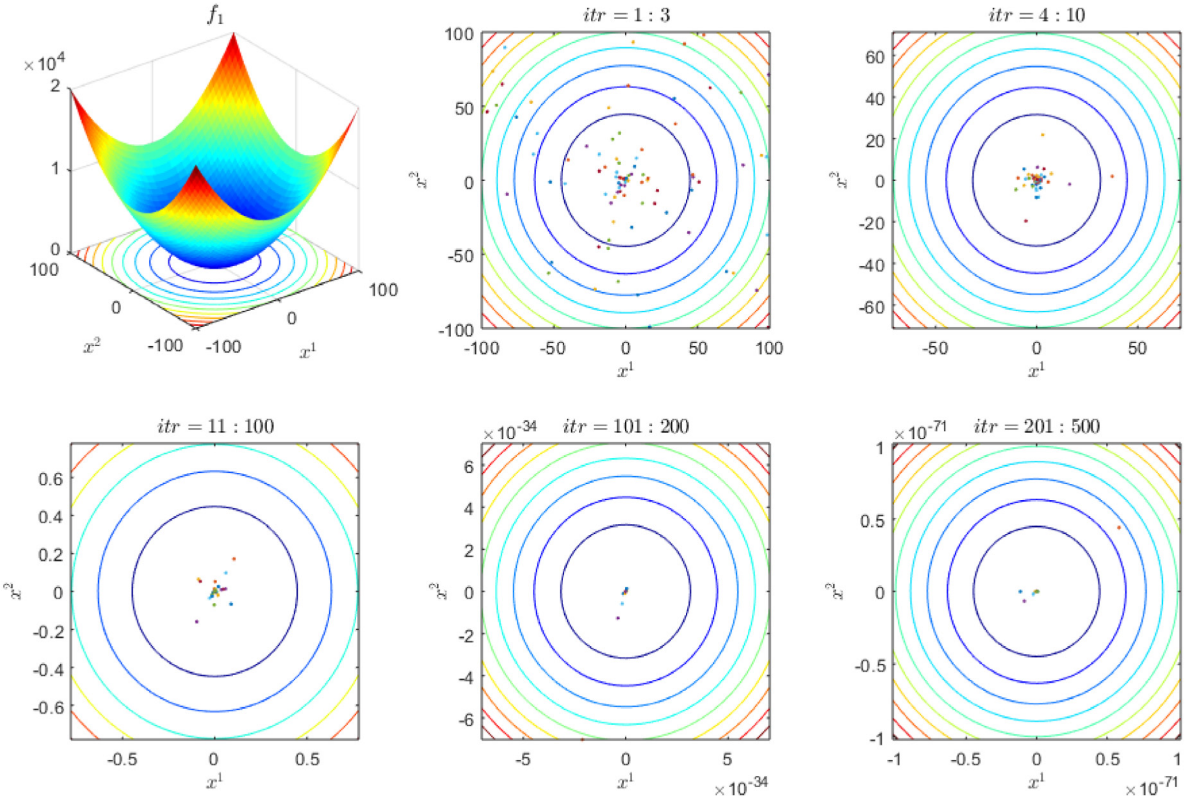

Fig. 2. Illustration of the searching pattern using the search history.

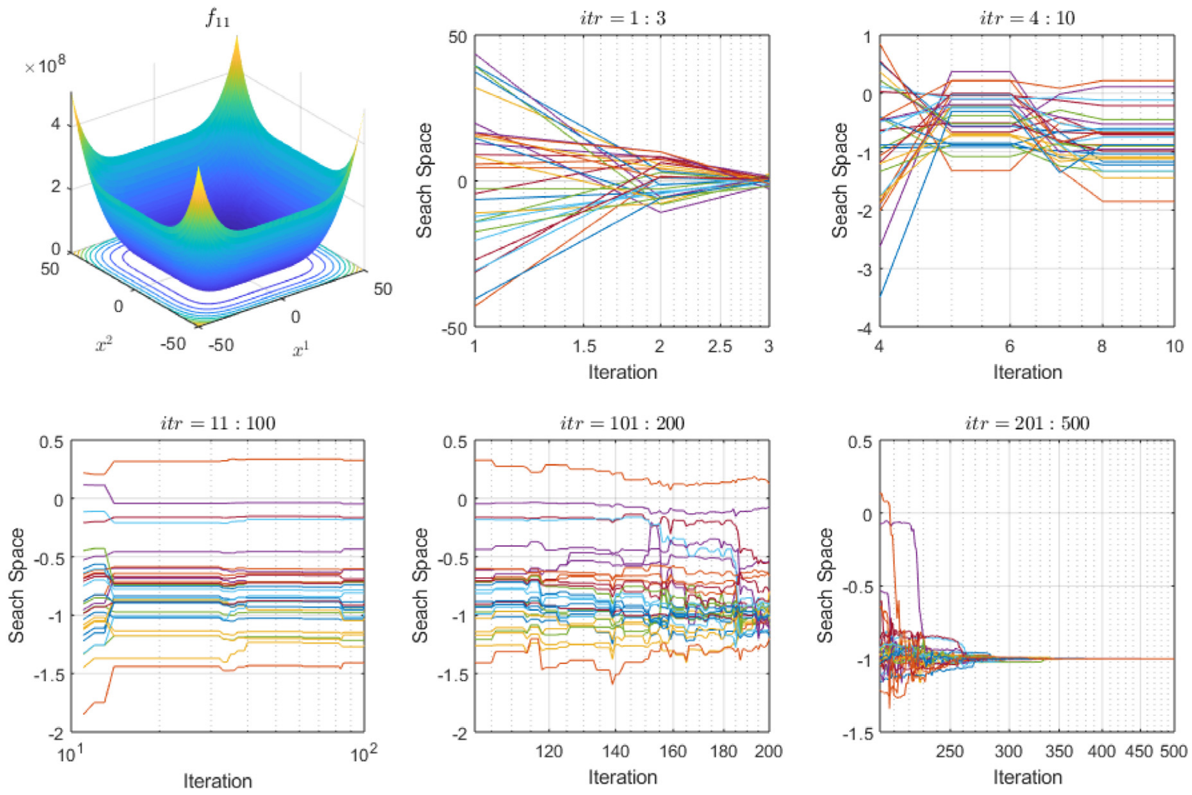

Fig. 3. Illustration of the Searching pattern using the trajectory of first Manta Ray.

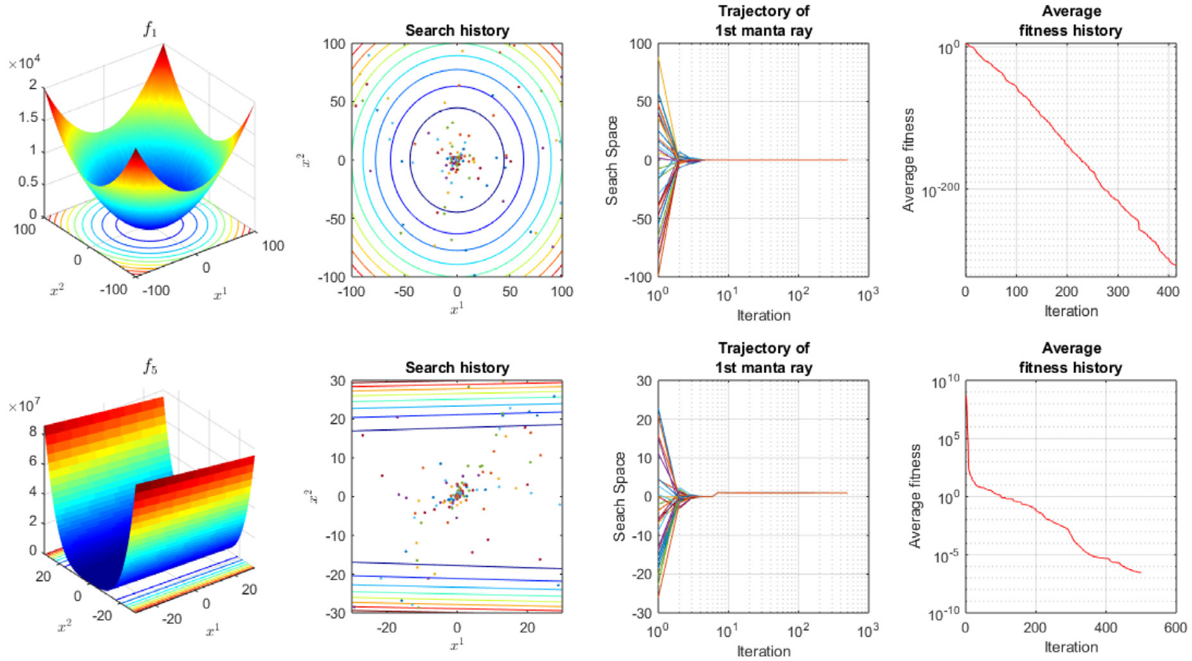

Fig. 4. Qualitative Results of unimodal test functions f_1 and f_5.

entropy has better noise resistance and edge preservation capability than 1D and 2D Tsallis Entropy.

The axes of a 3D histogram for a given image I of size $M\times N$ having L intensity levels $[0, L-1]$, represents the pixel intensity level $f(x,y)$, the local mean $f_{avg}(x,y)$ and median $f_{med}(x,y)$ values. The local mean and median value at a certain coordinate (x,y) in a $m\times m$ neighborhood region is expressed in Eqs. (1) and (2), respectively.

$$f_{avg}(x,y) = \frac{1}{m\times m}\sum_{u=-\frac{m-1}{2}}^{\frac{m-1}{2}}\sum_{v=-\frac{m-1}{2}}^{\frac{m-1}{2}} f(x+u,y+v) \qquad (1)$$

$$f_{med}(x,y) = med\left\{f(x+u,y+v);u,v=-\frac{m-1}{2}\dots\frac{m+1}{2}\right\} \qquad (2)$$

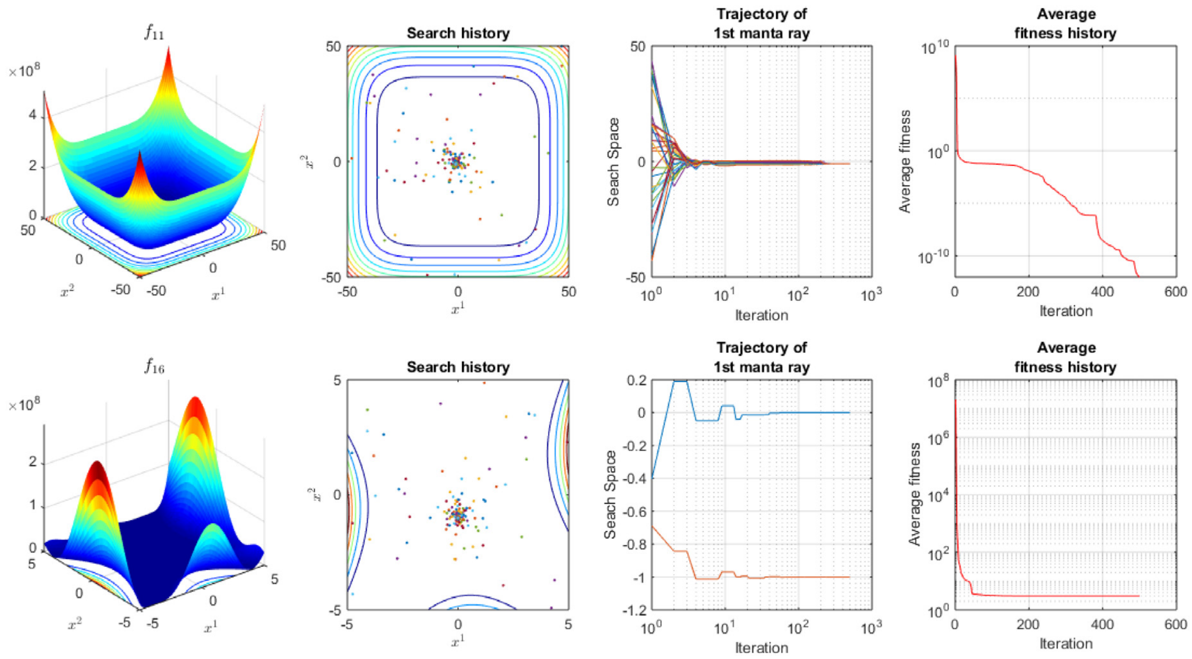

Fig. 5. Qualitative Results of multimodal test functions f_{11} and f_{16}.

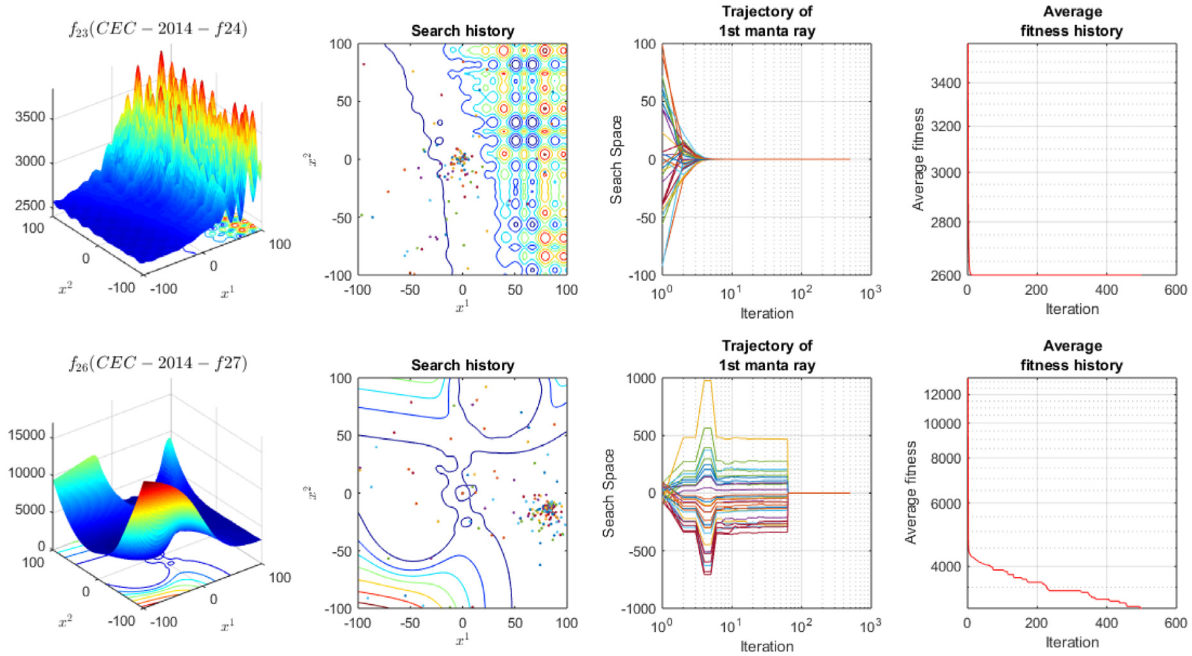

Fig. 6. Qualitative Results of composite test functions f_{23} and f_{26} from CEC 2014 test suit.

In this paper, the value of m is fixed and taken as 3 for all the images at a different level of segmentation. The pixel intensity $f(x,y) = i$ of an image with its corresponding mean $f_{avg}(x,y) = j$ and median $f_{med}(x,y) = k$ intensity values are combined to form a gray level triple (i,j,k). All possible triples in a 3D histogram are represented by its joint probability expressed in Eq. (3) within a cube of volume $L{\times}L{\times}L$ shown in Fig. 1(a)

$$p_{ijk} = \frac{\mu_{ijk}}{M \times N} \tag{3}$$

where μ_{ijk} is the number of occurrences of a triple (i,j,k) and $0 \leq i,j,k \leq L-1$.

Let us consider an arbitrary threshold point $(s_1,\ s_2; t_1, t_2; r_1, r_2)$ in the 3D histogram as shown in Fig. 1 for tri-level thresholding. s_1, s_2 denotes the threshold from the pixel gray levels, t_1, t_2 and r_1, r_2 are local mean and median thresholds, respectively. The given threshold points in tri-level thresholding split the 3D histogram into 27 distinct regions. Cube 27 and 14 in Figs. 1(b) and 1(c) considered as the two objects of interest and cube 1 in Fig. 1(d) represent the background. The remaining cubes generally contain information related to border pixels or noises. The probability of gray level triples in these regions are very less and approximated to zero, which can be ignored without affecting thresholding efficiency. The probability distribution $P_{27} and P_{14}$ of the above two object classes and background P_1 are given by

$$P_{27}(s,t,r) = P_{C1}(s,t,r) = \sum_{i=s_2+1}^{L-1} \sum_{j=t_2+1}^{L-1} \sum_{k=r_2+1}^{L-1} p_{ijk} \tag{4}$$

Table 2
Statistical results of benchmark functions.

	Function	Matric	AMRFO	MRFO	EO	HHO	SFO	GWO	PSO	DE
Unimodal test function	f_1	Ave	**0**	**0**	2.37E−41	5.00E−96	3.24E−15	6.56E−08	5.42E−02	1.37E+01
		Std	**0**	**0**	4.06E−41	2.79E−95	7.25E−15	1.62E−07	1.38E−02	5.36E+01
	f_2	Ave	3.63E−201	**1.39E−249**	4.24E−24	5.93E−50	1.45E−07	3.63E−05	1.07E+00	1.41E−02
		Std	**0**	**0**	3.96E−24	1.96E−49	1.75E−07	3.54E−05	1.58E−01	3.47E−02
	f_3	Ave	**0**	**0**	8.54E−08	1.33E−75	7.39E−13	2.26E−01	5.32E−01	5.92E+02
		Std	**0**	**0**	5.24E−07	9.43E−75	2.22E−12	2.75E−01	1.18E−01	4.24E+02
	f_4	Ave	2.51E−202	**3.43E−242**	3.20E−10	2.07E−47	9.17E−09	1.43E−01	1.43E−01	2.34E+01
		Std	**0**	**0**	7.65E−10	1.46E−46	1.31E−08	1.23E−01	2.18E−02	6.62E+00
	f_5	Ave	**2.49E−04**	2.30E+01	2.55E+01	1.13E−02	7.69E−02	2.87E+01	3.35E+01	8.22E+03
		Std	**4.85E−04**	4.82E−01	2.46E−01	1.84E−02	1.55E−01	3.29E−01	1.60E+00	2.89E+04
	f_6	Ave	**4.39E−10**	7.54E−10	1.11E−05	1.33E−04	3.01E+00	3.47E+00	5.14E−02	1.06E+01
		Std	**1.49E−09**	2.28E−09	7.99E−06	2.11E−04	1.81E+00	7.02E−01	1.36E−02	3.88E+01
	f_7	Ave	**1.06E−04**	1.10E−04	1.33E−03	1.34E−04	2.15E−04	5.52E−03	1.27E−01	6.76E−02
		Std	**7.78E−05**	9.80E−05	7.14E−04	1.26E−04	1.72E−04	3.34E−03	4.69E−02	1.65E−02
Scalable dimension Multimodal test function	f_8	Ave	**0**	**0**	1.95E−02	**0**	5.96E−13	3.18E+01	1.73E+01	1.57E+02
		Std	**0**	**0**	1.39E−01	**0**	1.47E−12	3.48E+01	4.37E+00	3.10E+01
	f_9	Ave	**8.88E−16**	**8.88E−16**	8.41E−15	**8.88E−16**	2.27E−08	5.51E−05	3.07E−01	2.15E+00
		Std	**0**	**0**	1.53E−15	**0**	2.51E−08	4.74E−05	1.31E−01	1.07E+00
	f_{10}	Ave	**0**	**0**	3.94E−04	**0**	1.11E−16	2.90E−09	3.36E−03	3.16E−01
		Std	**0**	**0**	1.97E−03	**0**	4.11E−16	4.76E−09	8.84E−04	4.45E−01
	f_{11}	Ave	**3.36E−10**	5.94E−10	2.03E−03	1.20E−05	5.47E−01	3.10E−01	7.58E−04	4.05E+03
		Std	**1.93E−09**	3.33E−09	1.45E−02	2.62E−05	2.66E−01	9.60E−02	2.39E−04	2.42E+04
	f_{12}	Ave	3.43E−03	2.00E+00	2.47E−02	**5.56E−05**	6.16E−03	2.11E+00	1.70E−02	3.50E+04
		Std	5.54E−03	1.33E+00	5.19E−02	**6.74E−05**	9.80E−03	2.79E−01	7.50E−03	1.26E+05
Fixed dimension Multimodal test function	f_{13}	Ave	7.17E−04	7.56E−04	2.37E−03	4.37E−04	6.58E−04	7.75E−04	**4.08E−04**	1.39E−03
		Std	4.18E−04	4.87E−04	6.00E−03	3.05E−04	1.85E−03	6.35E−04	**2.05E−04**	3.92E−03
	f_{14}	Ave	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0270**	**−1.0316**	**−1.0316**
		Std	2.88E−16	2.93E−16	3.08E−16	6.57E−09	3.79E−04	1.15E−02	1.27E−06	**2.31E−16**
	f_{15}	Ave	**0.3979**	**0.3979**	**0.3979**	**0.3979**	0.4095	0.5800	0.4889	**0.3979**
		Std	**3.92E−16**	**3.92E−16**	**3.92E−16**	2.79E−05	4.26E−02	9.10E−01	6.50E−01	**3.92E−16**
	f_{16}	Ave	**3**	**3**	**3**	**3**	7.7956	5.1407	3.0000	**3**
		Std	1.80E−15	2.40E−15	**1.40E−15**	1.80E−07	1.40E+01	7.32E+00	6.03E−05	4.18E−15
	f_{17}	Ave	**−3.8628**	**−3.8628**	−3.8626	−3.8602	−3.7931	−3.7407	−3.8565	**−3.8628**
		Std	**3.06E−15**	3.11E−15	1.10E−03	3.85E−03	7.43E−02	5.72E−01	3.12E−03	3.14E−15
	f_{18}	Ave	**−3.2964**	−3.2800	−3.2705	−3.0972	−2.9706	−2.4238	−3.1428	−3.2311
		Std	**4.94E−02**	5.74E−02	6.34E−02	1.06E−01	2.06E−01	7.80E−01	1.84E−01	5.09E−02
	f_{19}	Ave	**−10.1532**	−7.7115	−8.7653	−5.3295	−5.0109	−4.6455	−5.0548	−9.0720
		Std	**8.11E−15**	2.78E+00	2.59E+00	1.13E+00	5.49E−02	1.25E+00	3.51E−04	2.41E+00
	f_{20}	Ave	**−10.4029**	−8.0466	−9.9722	−5.2790	−5.3272	−4.5943	−5.0872	−9.5884
		Std	**9.51E−15**	3.05E+00	1.74E+00	9.76E−01	1.18E+00	1.36E+00	3.54E−04	2.27E+00
	f_{21}	Ave	**−10.5364**	−7.0232	−9.7507	−5.0695	−5.2762	−4.7984	−5.1281	−10.3861
		Std	**1.27E−14**	3.22E+00	2.22E+00	3.80E−01	9.04E−01	1.13E+00	3.43E−04	1.07E+00
Composition test function (CEC 2014)	f_{22}	Ave	**2500**	**2500**	2615.3165	**2500**	**2500**	2500.0055	2506.0424	2617.4092
		Std	**0**	**0**	7.37E−02	**0**	1.68E−06	7.52E−03	8.75E−01	8.01E+00
	f_{23}	Ave	**2600**	**2600**	2600.0235	2600.0003	2600.0005	2601.6712	2601.2768	2639.7510
		Std	**0**	**0**	1.13E−02	6.25E−04	4.36E−04	7.54E−01	1.59E−01	7.04E+00
	f_{24}	Ave	**2700**	**2700**	2701.4302	**2700**	2700.0000	2700.0001	2700.0950	2706.1518
		Std	**0**	**0**	4.00E+00	**0**	1.28E−08	6.19E−05	1.23E−02	1.42E+00
	f_{25}	Ave	**2700.6256**	2700.6399	2729.6636	2774.6299	2787.5090	2800.0003	2800.0017	2714.6172
		Std	**1.00E−01**	1.08E−01	4.59E+01	4.38E+01	3.16E+01	7.75E−04	4.03E−04	4.15E+01
	f_{26}	Ave	**2900**	2900.0001	3308.0067	**2900**	**2900**	2900.0007	2903.8909	3261.5505
		Std	**0**	9.82E−04	1.06E+02	**0**	2.05E−07	7.73E−04	9.04E−01	6.77E+01
	f_{27}	Ave	**3000**	**3000**	3815.4687	**3000**	3000.0000	3000.0017	3005.4869	3931.4169
		Std	**0**	**0**	1.55E+02	**0**	5.89E−07	1.64E−03	1.17E+00	2.40E+02
Friedman's mean rank			**1.89**	2.70	4.67	3.43	4.69	6.57	5.91	6.15
Rank			**1**	2	4	3	5	8	6	7

$$P_{14}\left(s,t,r\right) = P_{C2}\left(s,t,r\right) = \sum_{i=s_1+1}^{s_2} \sum_{j=t_1+1}^{t_2} \sum_{k=r_1+1}^{r_2} p_{ijk} \tag{5}$$

$$P_1\left(s,t,r\right) = P_{C3}\left(s,t,r\right) = \sum_{i=0}^{s_1} \sum_{j=0}^{t_1} \sum_{k=0}^{r_1} p_{ijk} \tag{6}$$

where $s = [s_1, s_2]$, $t = [t_1, t_2]$ and $r = [r_1, r_2]$

The 3D Tsallis entropy of each class distribution is expressed as

$$E_3^{\alpha}\left(s,t,r\right) = \frac{1}{\alpha - 1}\left[1 - \sum_{i=s_2+1}^{L-1} \sum_{j=t_2+1}^{L-1} \sum_{k=r_2+1}^{L-1} \left(\frac{p_{ijk}}{P_{C1}\left(s,t,r\right)}\right)^{\alpha}\right] \tag{7}$$

$$E_2^{\alpha}\left(s,t,r\right) = \frac{1}{\alpha - 1}\left[1 - \sum_{i=s_1+1}^{s_2} \sum_{j=t_1+1}^{t_2} \sum_{k=r_1+1}^{r_2} \left(\frac{p_{ijk}}{P_{C2}\left(s,t,r\right)}\right)^{\alpha}\right] \tag{8}$$

$$E_1^{\alpha}(s,t,r) = \frac{1}{\alpha - 1}\left[1 - \sum_{i=0}^{s_1}\sum_{j=0}^{t_1}\sum_{k=0}^{r_1}\left(\frac{p_{ijk}}{P_{C3}(s,t,r)}\right)^{\alpha}\right] \tag{9}$$

where α is the Tsallis entropy index (Sarkar and Das, 2013b).

The above entropy form can be extended for multilevel thresholding by a pseudo additive entropy rule as:

$$E^{\alpha}(s,t,r) = E_1^{\alpha} + E_2^{\alpha} + E_3^{\alpha} + (1-\alpha)\,E_1^{\alpha}E_2^{\alpha}E_3^{\alpha} \tag{10}$$

Hence the objective function for above tri-level thresholding to obtain optimal threshold (s^*, t^*, r^*) can be formulated as

$$\left(s^*, t^*, r^*\right) = \underset{0 \le s,t,r \le L-1}{\arg max}\ \left\{E^{\alpha}(s,t,r)\right\} \tag{11}$$

where, $s^* = [s_1^*, s_2^*]$, $t^* = [t_1^*, t_2^*]$ and $r^* = [r_1^*, r_2^*]$.

Once the optimal values (s^*, t^*, r^*) are obtained, the threshold vector contains two threshold values $\left[\frac{s_1^*+t_1^*+r_1^*}{3}, \frac{s_2^*+t_2^*+r_2^*}{3}\right]$ is used for generating the thresholded image.

The objective function can be extended to $K+1$-level thresholding for K thresholds values as

$$E^{\alpha}(s,t,r) = \arg max\left(E_1^{\alpha} + E_2^{\alpha} + \cdots + E_{K+1}^{\alpha} + (1-\alpha)\,E_1^{\alpha}E_2^{\alpha}\cdots E_{K+1}^{\alpha}\right) \tag{12}$$

where

$$E_n^{\alpha}(s,t,r) = \frac{1}{\alpha - 1}\left[1 - \sum_{i=s_{n-1}+1}^{s_n}\ \sum_{j=t_{n-1}+1}^{t_n}\ \sum_{k=r_{n-1}+1}^{r_n}\left(\frac{p_{ijk}}{P_{Cn}(s,t,r)}\right)^{\alpha}\right], \tag{13}$$

$$s = [s_1, s_2, \ldots, s_K],\ t = [t_1, t_2, \ldots, t_K]\ and\ r = [r_1, r_2, \ldots, r_K].$$

3. The proposed attacking Manta ray foraging optimization (AMRFO)

Manta ray foraging optimization (MRFO) (Zhao et al., 2020) imitates the foraging behaviors of one of the largest creatures found in the ocean known as Manta Rays. Manta Ray adopts three different kinds of foraging strategies as Chain foraging, cyclone foraging, and somersault foraging to reach food sources, which are planktons. In the Chain foraging strategy, more than 50 Manta Rays are involved and form a foraging chain by line up one after another. Manta Rays adopt cyclone foraging when the food concentration is high and move towards the food in a spiral-shaped path around it. Somersault foraging is a random cyclic movement of a Manta Ray around the current best position to optimize its foraging process. The development of the Attacking Manta ray foraging optimization algorithm (AMRFO) is based on the enrichment of energy levels in the foraging strategy of Manta Ray on exploration and exploitation stages of MRFO (Zhao et al., 2020) taking inspiration from sailfish optimizer (SFO) (Shadravan et al., 2019).

3.1. A mathematical formulation of AMRFO

AMRFO includes additional adaptive attacking power in different foraging strategy of Manta Ray, which not only improve the diversity of the population in the earlier stage but also prevent it to stick at any local minima or maxima at later stages. In this way, the algorithm maintains the desired balance between its initial exploration and later exploitation abilities. In each of the foraging behaviors discussed above a Manta Ray fish update its position by the current best solution and the solution in front of it. It has been observed that because of inadequate energy in the update algorithm of MRFO, it does not maintain the required diversity of the population to use the search space efficiently and escape from local minima or maxima once it is stuck there. Thus, the proposed algorithm includes an additional attacking power AP as given in (14), which updates itself concerning iterations and provides the required energy to overcome the above problems.

$$AP = P\cdot\left(1 - \frac{itr}{itr_{max}}\right)^{30\cdot\left(\frac{itr}{itr_{max}}\right)} \tag{14}$$

where P is the initial attacking power taken as 3 determined experimentally in this paper and allows AP to decrease from $P = 3$ to $P = 0$ when iterations progress. The AP triggers the Manta Ray to perform exploration efficiently in the initial stage of iterations due to supplementary energy and in later stages, it improves the exploitation capability to reach global optima. The $itranditr_{max}$ represent the current and maximum iterations, respectively. The positions of Manta Ray are updated using the chain, cyclone, and somersault foraging.

3.1.1. Chain foraging

In chain foraging, Manta Rays are in a chain during foraging when searching the plankton with high concentration. The first Manta Ray in this chain updates its position concerning the current best position, whereas the remaining Manta Rays update their position concerning the current best position as well as the position of the alee Manta Ray. This can be mathematically expressed with additional attacking power as

$$X_i(itr+1) = \begin{cases} X_i(itr) + r_1\cdot\left(X_{best} - X_i(itr)\right) \\ \quad +\gamma\cdot\left(X_{best} - X_i(itr)\right) + r_2\cdot AP & i = 1 \\ X_i(itr) + r_1\cdot\left(X_{i-1}(itr) - X_i(itr)\right) \\ \quad +\gamma\cdot\left(X_{best} - X_i(itr)\right) + r_2\cdot AP & i = 2,3,\ldots,NP \end{cases} \tag{15}$$

where, $X_i(itr+1)$ and $X_i(itr)$ are the next position vector and the current position vector of i^{th} population. X_{best} is the current best position and $X_{i-1}(itr)$ is the $(i-1)^{th}$ individual in the population. r_1 is a random vector in the range $[0,1]$, r_2 is a random number in the range $[0,1]$ The parameter γ is known as weight co-efficient, which is calculated as

$$\gamma = 2r\cdot\sqrt{|\log(r)|} \tag{16}$$

where r is taken as a random vector within the range $[0,1]$.

3.1.2. Cyclone foraging

In cyclone foraging the Manta Rays present in a chain follows a spiral path towards the food during foraging. The updating rule for cyclone foraging with supplementary attacking power can be written as

$$X_i(itr+1) = \begin{cases} X_{best}(itr) + r_1\cdot\left(X_{best} - X_i(itr)\right) \\ \quad +\beta\cdot\left(X_{best} - X_i(itr)\right) + r_2\cdot AP & i = 1 \\ X_{best}(itr) + r_1\cdot\left(X_{i-1}(itr) - X_i(itr)\right) \\ \quad +\beta\cdot\left(X_{best} - X_i(itr)\right) + r_2\cdot AP & i = 2,3,\ldots,NP \end{cases} \tag{17}$$

where β is the weight co-efficient used for controlling the spiral movement and calculated as

$$\beta = 2e^{r_1\frac{itr_{max}-itr+1}{itr_{max}}}\cdot\sin\left(2\pi r_1\right) \tag{18}$$

where, r_1 is a random number in range $[0,1]$. The spiral movement is excellent for the exploitation approach, but to overcome the trap in local minima an exploration along with exploitation is required during iterations. This can be achieved by the current position of Manta Ray along with the randomly chosen Manta ray from a population in spiral foraging, which can be modeled as:

$$X_i(itr+1) = \begin{cases} X_{rand}(itr) + r_1\cdot\left(X_{rand} - X_i(itr)\right) \\ \quad +\beta\cdot\left(X_{rand} - X_i(itr)\right) + r_2\cdot AP & i = 1 \\ X_{rand}(itr) + r_1\cdot\left(X_{i-1}(itr) - X_i(itr)\right) \\ \quad +\beta\cdot\left(X_{rand} - X_i(itr)\right) + r_2\cdot AP & i = 2,3,\ldots,NP \end{cases} \tag{19}$$

where, X_{rand} is any arbitrary random position within the search space.

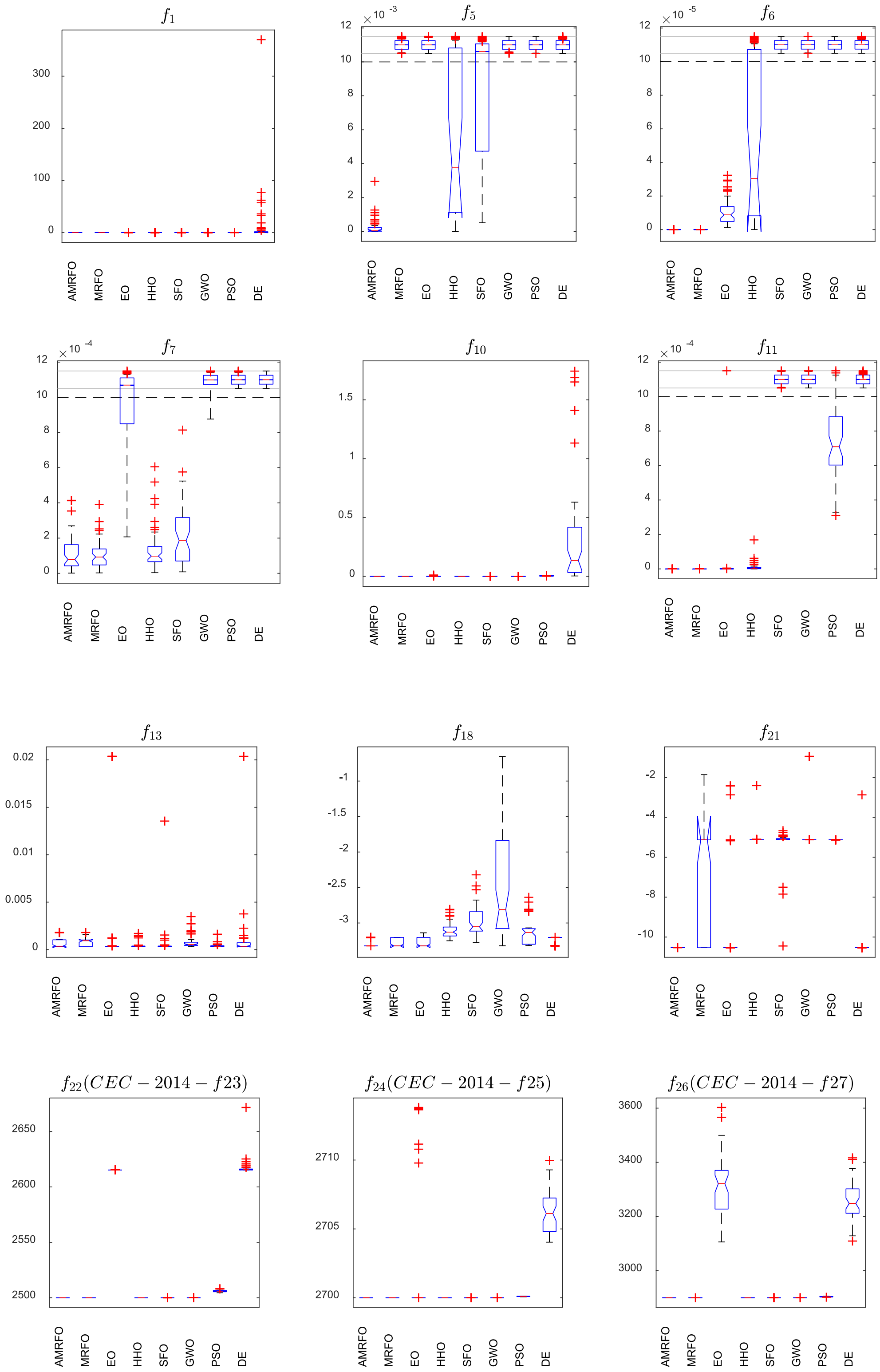

Fig. 7. Box plot of six unimodal, three multimodal, and three composite functions.

3.1.3. Somersault foraging

In somersault foraging, the best position found so far is considered nutrient-rich and all the individuals can adapt their position by a swim around it and somersault to a new location. This helps the algorithm to discover a new position between the current location and its symmetrical location around the nutrient rich. Then, the somersault foraging behavior with supplementary power can be expressed as

$$X_i(itr+1) = X_i(itr) + S \cdot \left(r_1 \cdot X_{best} - r_3 \cdot X_i(itr)\right) + r_2 \cdot AP \qquad (20)$$

where S is the Somersault factor indicates the somersault range, r_1 and r_3 are the random numbers in the range $[0, 1]$.

Table 3
p-values of Wilcoxon's signed-rank test for AMRFO against seven well-known algorithms with 50 independent runs. Counts of (+)/(≈)/(−) represent the number of cases AMRFO is statistically superior/similar/inferior than/to other algorithms with a level of significance ($\alpha = 0.05$).

Test functions	AMRFO vs. MRFO	AMRFO vs. EO	AMRFO vs. HHO	AMRFO vs. SFO	AMRFO vs. GWO	AMRFO vs. PSO	AMRFO vs. DE
f_1	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_2	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_3	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_4	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_5	8.88E−16	8.88E−16	2.33E−09	4.62E−14	8.88E−16	8.88E−16	8.88E−16
f_6	4.60E−03	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_7	**7.80E−01**	8.88E−16	**9.19E−02**	4.89E−02	8.88E−16	8.88E−16	8.88E−16
f_8	**1.00E+00**	**1.00E+00**	**1.00E+00**	2.33E−10	8.88E−16	8.88E−16	8.88E−16
f_9	**1.00E+00**	8.88E−16	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{10}	**1.00E+00**	**2.50E−01**	**1.00E+00**	9.77E−04	8.88E−16	8.88E−16	8.88E−16
f_{11}	4.60E−03	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{12}	8.88E−16	5.70E−05	4.60E−03	1.77E−03	8.88E−16	1.97E−11	4.62E−14
f_{13}	**7.80E−01**	**5.76E−01**	**5.76E−01**	**5.76E−01**	**5.76E−01**	**1.00E+00**	4.89E−02
f_{14}	**1.00E+00**	**6.90E−01**	8.88E−16	8.88E−16	8.88E−16	8.88E−16	9.77E−04
f_{15}	**1.00E+00**	**1.00E+00**	7.11E−15	8.88E−16	8.88E−16	8.88E−16	**1.00E+00**
f_{16}	**1.35E−01**	2.26E−02	4.62E−14	8.88E−16	8.88E−16	8.88E−16	1.46E−10
f_{17}	**6.54E−02**	2.27E−02	8.88E−16	8.88E−16	8.88E−16	8.88E−16	1.95E−03
f_{18}	**5.25E−02**	3.51E−02	1.18E−12	8.88E−16	4.62E−14	1.83E−08	3.47E−06
f_{19}	2.94E−03	7.66E−07	8.88E−16	8.88E−16	8.88E−16	8.88E−16	6.80E−04
f_{20}	3.51E−02	2.77E−04	8.88E−16	8.88E−16	8.88E−16	8.88E−16	**2.86E−01**
f_{21}	4.87E−06	2.32E−03	8.88E−16	8.88E−16	8.88E−16	8.88E−16	4.92E−05
f_{22}	**1.00E+00**	8.88E−16	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{23}	**1.00E+00**	8.88E−16	5.68E−14	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{24}	**1.00E+00**	8.88E−16	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{25}	**4.01E−01**	4.89E−02	5.70E−05	8.88E−16	8.88E−16	8.88E−16	**7.80E−01**
f_{26}	**1.00E+00**	8.88E−16	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16
f_{27}	**1.00E+00**	8.88E−16	**1.00E+00**	8.88E−16	8.88E−16	8.88E−16	8.88E−16
Count: (+)/(≈)/(−)	9/12/6	22/2/3	18/7/2	26/0/1	26/0/1	26/1/0	24/1/2

3.2. Pseudocode of AMRFO algorithm

Let us initialize the size of the population, represents the number of Manta Ray take part in searching for an optimal solution as NP and its initial position as $X_{i,} = [x_i^1, x_i^2, \dots, x_i^D], where i = 1, 2, \dots, NP$ and D denotes the problem dimension, the boundary of the search space using upper bound $X_{ub} = 255$ and lower bound $X_{lb} = 0$, the maximum number of iterations as itr_{max} and the objective function $F(X_i)$.

4. Performance evaluation of AMRFO algorithm

To evaluate the performance of the proposed AMRFO algorithm, tests have been carried out by a set of 21 diverse classical benchmark functions (($f_1 - f_{21}$) taken from the literature (Wunnava et al., 2020b). Out of these 21 benchmark functions, ($f_1 - f_7$) are unimodal functions with unique global minima used to validate the exploitative capability of the optimization algorithm. Test using multimodal benchmark functions ($f_8 - f_{12}$) with scalable dimension and ($f_{13} - f_{21}$) with a fixed dimension having many local minima are used to observe how effectively to avoid local minima during exploration. As the real-world problems have no specific structure, 6 complex composite functions ($f_{22} - f_{27}$) from CEC2014 test suits (Liang et al., 2013) also have been used to reveal the performance of the algorithm. The details of the benchmark functions are given in Appendix.

To validate the efficacy of the proposed algorithm, the performance results of AMFRO are compared with well-known nature-inspired optimization algorithms such as MRFO (Zhao et al., 2020), EO (Abdel-Basset et al., 2020), HHO (Heidari et al., 2019), SFO (Shadravan et al., 2019), GWO (Khairuzzaman and Chaudhury, 2017), PSO (Maitra and Chatterjee, 2008b) and DE (Sarkar and Das, 2013a). The average ('Ave') and standard deviation ('Std') values are taken as a quantitative measure based on 51 independent runs of each benchmark test function, whereas qualitative measure includes search history, trajectory, average fitness history, box plot, convergence curve, and scalability analysis. To effectively detect statistical differences between optimization algorithms, most used the nonparametric Wilcoxon signed-rank test at a 5% significance level ($\alpha = 0.05$) is also performed. This test shows how better or inferior an optimization algorithm is compared with another algorithm using p value. A '+' sign represents the proposed algorithm conquers over the compared algorithm and vice versa for the '−' sign. Another nonparametric test is known as Friedman's mean rank also used to rank the algorithm by observing its average performance in all benchmark test functions. Rank 1 is allotted to the best-performing algorithm.

The population size (NP) and maximum number of iterations (itr_{max}) for all considered optimization algorithms are taken as 30 and 500, respectively. As the initial parameter setting of different optimizers to produce satisfactory results are required, which is presented in Table 1. All the experiments are done in (MATLAB 2015) environment in the i3-intel core processor (8th gen.) with a RAM capacity of 8 GB.

4.1. Qualitative analysis of AMRFO

Results suitable for qualitative analysis of AMFRO are shown in Figs. 2–6. It includes three qualitative metrics of six different test functions: search history, the trajectory of the first Manta Ray, and the average fitness value. These clearly show the search pattern of Manta Ray and in what way a Manta Ray contributes to getting the optimal solution in the proposed AMRFO. For a better illustration of the search patterns using the search history, the variations of the positions (colorful dots) of NP Manta Rays throughout iterations are presented in Fig. 2. It can be observed that initially the Manta Rays explore the promising area of the 2D search space and start exploiting around the global optima when iterations increase. The trajectory map of the first Manta Ray in Fig. 3 shows how the 30 different control variables of a Manta Ray vector in a 2D space varies in the process phases when the number of iterations increase. Each line in the figure represents the trajectory of a control variable from the beginning to the end of the iteration. In the initial state of the process, the searching process starts from random locations and the variation of their movement is high enough to reach most of the promising areas in the search space. When time progresses, the magnitude of the variation decreases, and the algorithm makes a transition from exploration state to exploitation state. Finally, the movement of the first Manta Ray becomes almost stable, indicating that the algorithm is about to reach the global optima.

```
Begin
Inputs: $NP, X_{ub}$ , $X_{lb}$ and $itr_{max}$.
Initialization: $X = [X_{1,}\ X_2, \cdots, X_{NP}\ ]$, with rand position, where $X_i = X_{lb} + rand() \cdot (X_{ub} - X_{lb})$.
Set the initial iteration: $itr = 1$.
While $itr \leq itr_{max}$
  I.   Evaluate the objective function $F(X_i)$ for all initial positions of Manta Ray.
  II.  Based on the best objective function, obtain the current best position of Manta Ray and name as $X_{best}$.
  III. Update the Attacking power using Eq. (14).
  IV.  For $i = 1$ to $NP$ do
       # For $i$th Manta Ray
           If $rand() < 0.5$
               # Select cyclone foraging
               if $\frac{itr}{itr_{max}} < rand()$
                       Update the position $X_i$ of $i$th Manta Ray using Eq. (19).
               Else
                       Update the position $X_i$ of $i$th Manta Ray using Eq. (17).
               End if
             Else
             #Select Chain foraging
               Update the position $X_i$ of $i$th Manta Ray using Eq. (15).
           End if
       Compare the fitness of new positions of $i$th Manta Ray $X_i$ with fitness of $X_{best}$ and update $X_{best}$ if
       required.
       end for
  V.   For $i = 1$ to $NP$ do
       # Perform Somersault foraging
         # For $i$th Manta Ray
         Update the position $X_i$ of $i$th Manta Ray using Eq. (20).
         Evaluate the objective function modified position and update the best position of Manta Ray $X_{best}$ if
         required.
       end for
  VI.  $itr = itr + 1$
        End While
  VII.  Return the optimal position of Manta Ray as $X_{best}$.
Output: Optimal position $X_{best}$.
```

The average fitness value represents the variation of the average fitness value throughout the optimization process.

Figs. 4, 5 and 6 show the qualitative metrics of unimodal, multimodal, and composite test functions, respectively. The second column of these figures shows the search history of all the Manta Rays in the population at different steps of the optimization process. The search history plots in these figures are generated by accumulating the search locations obtained in different stages as shown in Fig. 2. It can be observed from the search history that the maximum number of Manta Rays are concentrated in the desired minima on the contour plot of $f_1, f_5, f_{11}, f_{16}, f_{24}$ and f_{27}. From the Trajectory diagram of f_1, f_5, f_{11} and f_{16}, it is visible that in the initial state of the iteration, the curve relates to the position of first Manta Ray oscillates due to performing explorations and later converges to an optimal location. Due to the more complex behavior of composite functions f_{24}, and f_{27}, the trajectory diagram includes more fluctuation in early states of iterations and then converges. To observe the collaborative behavior of Manta Rays, the average fitness history of different test functions over 500 iterations is presented in the fourth column. The decreasing graph in all the considered cases shows that, all the Manta Ray participated in the foraging simultaneously updated their current position by moving to a better location.

4.2. AMRFO performance on benchmark functions

The statistical results of classical unimodal functions ($f_1 - f_7$), multimodal functions of scalable and fixed dimensions ($f_8 - f_{21}$) and composite functions of modern CEC 2014 test suits ($f_{22} - f_{27}$) along with the result of Friedman's mean rank test are presented in Table 2. The best values in the table for every benchmark function are highlighted in bold font. It can be observed from the Friedman test that, AMRFO was placed in Rank 1 among all other well-known algorithms. Even though AMRFO lags from another optimization algorithm in test functions $f_2, f_4, f_{12}, f_{13}, f_{14}$, and f_{16}, it dominates the others in most cases. Compared with MRFO, AMRFO outperforms more than half of the test functions used for performance evaluation. It can also be seen that none of the compared algorithms including MRFO could find the global optima for f_5, but the solution ability of AMRFO is much better than others. AMRFO proofs its exploitation capability by outperforming most of the unimodal functions. To validate the exploration ability

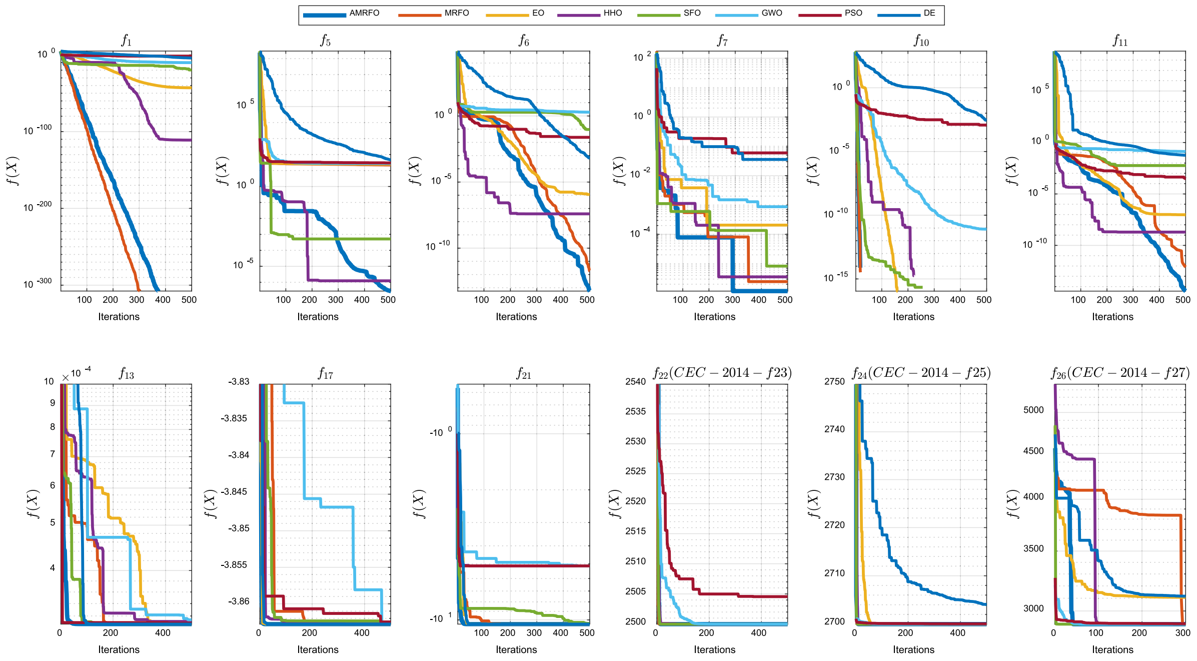

Fig. 8. Convergence curve of six unimodal, three multimodal and three composite functions.

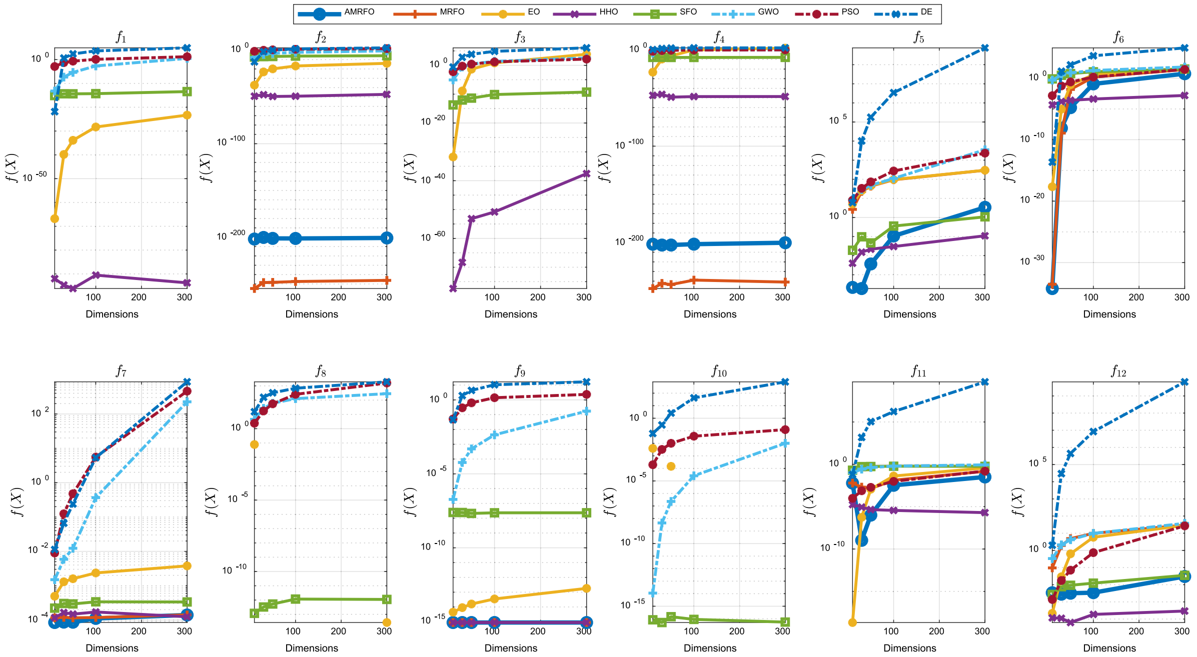

Fig. 9. Scalability results comparison of AMRFO to other well-known algorithms of classical test functions.

of the proposed AMRFO, the results of multimodal functions having multiple minima must be taken into consideration. The recorded result under this multimodal category shows that AMRFO is superior on $f_{11}, f_{17}, f_{18}, f_{19}, f_{20}$, and f_{21} whereas comparable results on f_{14} and f_{16} with DE and EO, respectively. PSO and HHO dominates all methods in f_{12} and f_{13} solution respectively by its superior performance. However, the exploration capability of AMFRO is satisfactory as compared to others. Results of composite functions show that AMFRO can be applied to solve real-world complex problems. A pictorial representation of the results of 12 benchmark test functions collected from 51 independent runs of each algorithm is presented as a Box plot in Fig. 7. Boxplot (Williamson et al., 1989) is a standard way to view the distribution of data based on minimum, first quartile, median, third quartile, and maximum values. It can say, how well the data is lumped together. It can be observed from the boxplots, that the first quartile and third quartile which are 25th and 75th percentile of AMRFO samples decreased towards the lowest solution within a narrow interquartile range for all most all the test functions. The performance of AMRFO is thus satisfactory in comparison with other methods.

For further analysis of the performance of AMRFO, p-value of the non-parametric Wilcoxon signed-rank test at a 95% significance level of all test functions are listed in Table 3. The p-values less than 0.05 are considered as a win (+ sign) of the proposed method against the compared algorithms. A loss (− sign) of the proposed method represented by p-values scored more than 0.05, p-values equal to 1 denote that proposed and compared algorithms have identical performance

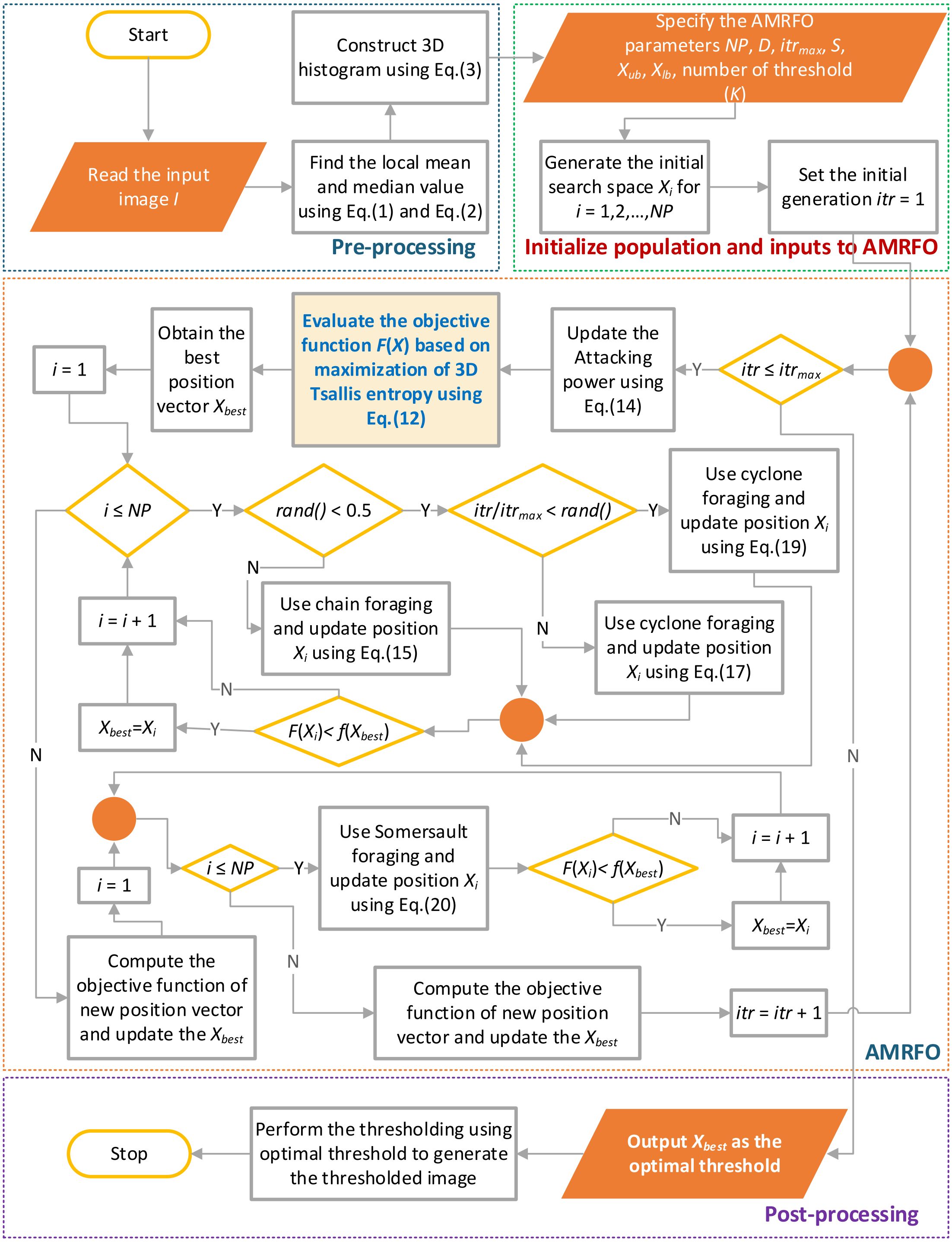
Start
Construct 3D histogram using Eq.(3)
Read the input image I
Find the local mean and median value using Eq.(1) and Eq.(2)
Pre-processing
Specify the AMRFO parameters NP, D, itr_{max}, S, X_{ub}, X_{lb}, number of threshold (K)
Generate the initial search space X_i for i = 1,2,…,NP
Set the initial generation itr = 1
Initialize population and inputs to AMRFO
i = 1
Obtain the best position vector X_{best}
Evaluate the objective function $F(X)$ based on maximization of 3D Tsallis entropy using Eq.(12)
Update the Attacking power using Eq.(14)
Y
$itr \leq itr_{max}$
$i \leq NP$
Y
$rand() < 0.5$
Y
$itr/itr_{max} < rand()$
Y
Use cyclone foraging and update position X_i using Eq.(19)
N
$i = i + 1$
Use chain foraging and update position X_i using Eq.(15)
N
Use cyclone foraging and update position X_i using Eq.(17)
N
X_{best}=X_i
Y
$F(X_i)< f(X_{best})$
N
N
N
$i = i + 1$
$i \leq NP$
Y
Use Somersault foraging and update position X_i using Eq.(20)
$F(X_i)< f(X_{best})$
i = 1
Y
X_{best}=X_i
N
Compute the objective function of new position vector and update the X_{best}
Compute the objective function of new position vector and update the X_{best}
$itr = itr + 1$
AMRFO
Stop
Perform the thresholding using optimal threshold to generate the thresholded image
Output X_{best} as the optimal threshold
Post-processing

Fig. 10. Flow chart of Maximum 3D Tsallis based multilevel thresholding using AMRFO.

and marked as similar (≈ sign). From the last row of Table 3, it can be observed that the proposed method scored more '+' signs against all algorithms. Therefore, AMRFO performance can be considered superior compared to seven algorithms in the Wilcoxon signed-rank test.

Convergence curves of the different algorithms for 12 benchmark test functions of different categories are shown in Fig. 8. The curve depicts the best fitness obtained by an algorithm so far vs iterations. Based on convergence capability, AMRFO dominates all algorithms for test functions $f_5, f_6, f_7, f_1, f_{17}$, and f_{21}. From the convergence curves of composite functions f_{23} and f_{27} given in the figure, it is can be observed that, AMRFO has the same converging rate as most of the well-known algorithms. Overall, AMRFO can be declared as a suitable algorithm for optimization problems.

To show how robust is the proposed AMRFO algorithm for a low to a high dimensional problem, the scalability test is also performed, and the results are depicted in Fig. 9. The dimensions are taken as $D = 10$, 30, 50, 100, and 300 with a population size of 30 for the scalability test. Like other qualitative analyses, here also the proposed algorithm compared with well-known algorithms with 12 classical benchmark test functions. The AMRFO beat the 7 compared algorithms for the test functions f_1, f_3, f_7, f_8, f_9, and f_{10} in the scalability test, whereas gives satisfactory performance for test functions f_3, f_5 and f_6 by lagging closely from its competitor.

Table 4
Performance measures.

Performance Measure	Formulation	Description	Reference
Peak Signal to Noise Ratio ($PSNR$)	$PSNR(dB) = 10\log 10\left(\frac{255^2}{MSE}\right)$ where Mean Square Error (MSE) for an $M \times N$ is defined as $MSE = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(X_{i,j}-\hat{X}_{i,j})^2$ where, $X_{i,j}$ represents the reference image and $\hat{X}_{i,j}$ is the test image.	PSNR is the most common measure for computing the quality of any test image concerning a reference image. For an 8-bit image, it is the ratio between maximum power (255^2) and the Mean square error value between two images. Higher the PSNR value represents better the quality of the test image.	Agrawal et al. (2013)
Feature Similarity Index ($FSIM$)	$FSIM = \frac{\sum_{x\in\Omega} S_L(x).PC_m(x)}{\sum_{x\in\Omega} PC_m(x)}$ where, $PC_m(x)$ is the maximum of the Phase Congruency map between test and reference image. $S_L(x)$. Is the product of similarity measure of Gradient Magnitude map and Phase Congruency map.	FSIM evaluates the importance of the local structure between the reference and test image. The maximum FSIM value that can be attended is 1.	Lin Zhang (2011)
Performance Measures Structural Similarity Index Measure ($SSIM$)	$SSIM = \frac{(2\mu_X\mu_{\hat{X}}+C_1)(2\sigma_{X\hat{X}}+C_2)}{(\mu_X^2+\mu_{\hat{X}}^2+C_1)(\sigma_X^2+\sigma_{\hat{X}}^2+C_2)}$ where, μ and σ are mean and standard deviation, respectively. C is the contrast comparison	The structural similarity measure is a function of luminance comparison, contrast comparison, and structure comparison of test and reference image. The closer the value of SSIM to 1 represents better the segmentation result	Daoud et al. (2017)

Table 5
Comparison of different optimization algorithms (Computed over 100 test images from AANLIB MR image dataset) for threshold level $(K) = 2$, 3, 4, and 5 using average fitness values and standard deviations.

K	Algorithm	3D Tsallis (3DTE)		2D Tsallis (2DTE)		1D Tsallis (1DTE)	
		f_{Avg}	std	f_{Avg}	std	f_{Avg}	std
2	AMFRO	**4.75E+09**	**1.89471E+09**	**2.63E+08**	**1.25743E+06**	**2.29E+04**	**2.00702E+03**
	MFRO	4.75E+09	1.94960E+09	2.63E+08	1.25743E+06	2.29E+04	2.00702E+03
	EO	4.74E+09	1.95149E+09	2.63E+08	1.05037E+08	2.29E+04	2.00702E+03
	HHO	4.75E+09	1.95149E+09	2.63E+08	9.84830E+07	2.29E+04	2.00702E+03
	SFO	4.71E+09	1.95228E+09	2.63E+08	1.05090E+08	2.29E+04	2.00702E+03
	GWO	4.01E+09	2.21746E+09	2.57E+08	1.05233E+08	2.29E+04	2.00702E+03
	PSO	4.47E+09	1.96041E+09	2.63E+08	1.05098E+08	2.29E+04	2.00702E+03
	DE	4.38E+09	2.02091E+09	2.62E+08	1.05118E+08	2.29E+04	2.00702E+03
3	AMFRO	**2.33E+12**	**7.39416E+11**	**4.91E+10**	**1.01678E+08**	**2.34E+05**	**2.58689E+04**
	MFRO	2.28E+12	1.05574E+12	4.91E+10	1.01678E+08	2.34E+05	2.59010E+04
	EO	2.22E+12	1.06289E+12	4.90E+10	2.46997E+10	2.34E+05	2.59010E+04
	HHO	2.23E+12	1.05805E+12	4.91E+10	2.26168E+10	2.34E+05	2.59010E+04
	SFO	2.20E+12	1.06289E+12	4.88E+10	2.49953E+10	2.34E+05	2.59010E+04
	GWO	1.57E+12	1.31132E+12	4.55E+10	2.50557E+10	2.34E+05	2.60014E+04
	PSO	2.07E+12	1.06318E+12	4.62E+10	2.50020E+10	2.34E+05	2.59010E+04
	DE	1.88E+12	1.11101E+12	4.60E+10	2.50493E+10	2.34E+05	2.59010E+04
4	AMFRO	**7.56E+14**	**1.87838E+14**	**6.39E+12**	**5.75822E+09**	**1.87E+06**	**2.33885E+05**
	MFRO	7.36E+14	3.15858E+14	6.38E+12	5.75822E+09	1.87E+06	2.36600E+05
	EO	6.00E+14	3.30822E+14	6.35E+12	3.52280E+12	1.87E+06	2.38679E+05
	HHO	6.98E+14	3.15858E+14	6.35E+12	2.79056E+12	1.87E+06	2.38555E+05
	SFO	5.45E+14	3.79241E+14	6.29E+12	3.91770E+12	1.87E+06	2.38691E+05
	GWO	3.32E+14	4.60855E+14	4.36E+12	3.93841E+12	1.84E+06	2.39196E+05
	PSO	5.33E+14	4.26655E+14	5.58E+12	3.92754E+12	1.87E+06	2.38806E+05
	DE	4.73E+14	4.59551E+14	4.49E+12	3.93614E+12	1.86E+06	2.39196E+05
5	AMFRO	**1.76E+17**	**5.56645E+16**	**6.03E+14**	**1.89100E+11**	**1.25E+07**	**1.82325E+06**
	MFRO	1.64E+17	5.56645E+16	6.03E+14	1.89100E+11	1.25E+07	1.82325E+06
	EO	1.59E+17	8.66729E+16	5.91E+14	3.40410E+14	1.25E+07	1.88253E+06
	HHO	1.62E+17	6.36216E+16	5.94E+14	2.98182E+14	1.25E+07	1.88087E+06
	SFO	1.06E+17	1.04646E+17	5.83E+14	4.27392E+14	1.25E+07	1.88321E+06
	GWO	8.09E+16	1.25286E+17	3.14E+14	4.30142E+14	1.19E+07	1.93714E+06
	PSO	1.05E+17	1.06276E+17	4.31E+14	4.28881E+14	1.21E+07	1.88387E+06
	DE	8.20E+16	1.09035E+17	3.59E+14	4.28980E+14	1.21E+07	1.88426E+06

5. A framework of the proposed 3D Tsallis entropy-based multilevel thresholding using AMRFO

In this section, we introduce the maximum 3D Tsallis entropy-based multilevel thresholding using the AMRFO algorithm to decompose the test images into $K+1$ distinct regions with the help of K optimal threshold values. As discussed in Section 2, 3D Tsallis entropy is used as an objective function to generate the optimal threshold matrix in terms of triple vector (s, t, r), which represents the threshold values correspond to the intensity of original, mean, and median images, respectively. The mean and median images are generated by considering intensity values within the 3×3 region around a pixel. In multilevel thresholding, each variable in the matrix represents a threshold vector containing a set of real numbers of $[0, L-1]$, where $L-1$ represents the maximum possible intensity value, which is further rounded off to the nearest integer. The threshold vectors s, t, and r for which the objective function attained the maximum values are referred to as optimal threshold vectors. The procedure of obtaining optimal threshold values by maximizing the 3D Tsallis entropy is an optimization problem, which can be done effectively with the help of a suitable optimization algorithm.

Table 6
Comparison of different optimization algorithms (Computed over 100 test images from AANLIB MR image dataset) for threshold level $(K) = 2$, 3, 4, and 5 using average PSNR, FSIM, and SSIM values.

K	Algorithm	3D Tsallis (3DTE)			2D Tsallis (2DTE)			1D Tsallis (1DTE)		
		$PSNR_{Avg}$	$FSIM_{Avg}$	$SSIM_{Avg}$	$PSNR_{Avg}$	$FSIM_{Avg}$	$SSIM_{Avg}$	$PSNR_{Avg}$	$FSIM_{Avg}$	$SSIM_{Avg}$
2	AMFRO	**23.0882**	**0.7087**	**0.4069**	22.8061	0.6982	0.3718	19.2997	0.6268	0.3000
	MFRO	23.0249	0.7061	0.4068	22.6209	0.6843	0.3602	19.2732	0.6268	0.2956
	EO	22.8830	0.7060	0.4041	21.9301	0.6839	0.3600	19.2526	0.6265	0.2950
	HHO	23.0249	0.7061	0.4068	21.9311	0.6842	0.3602	19.2628	0.6265	0.2950
	SFO	22.8515	0.7039	0.4029	21.9233	0.6834	0.3595	19.2365	0.6261	0.2948
	GWO	21.8983	0.6906	0.3900	21.7401	0.6812	0.3558	19.2223	0.6249	0.2947
	PSO	22.6637	0.7013	0.3979	21.9028	0.6834	0.3561	19.2366	0.6256	0.2948
	DE	22.2755	0.7008	0.3954	21.7534	0.6830	0.3560	19.2314	0.6251	0.2948
3	AMFRO	**26.2380**	**0.7993**	**0.5596**	26.1134	0.7797	0.5271	24.2047	0.7378	0.4364
	MFRO	26.1134	0.7977	0.5577	25.7056	0.7777	0.5144	24.1524	0.7378	0.4328
	EO	25.9781	0.7962	0.5474	25.5459	0.7768	0.5009	24.0885	0.7364	0.4274
	HHO	26.0388	0.7977	0.5532	25.5787	0.7775	0.5096	24.1327	0.7377	0.4309
	SFO	25.87237	0.7939	0.5356	25.442	0.7766	0.5004	24.0885	0.7360	0.4252
	GWO	25.601	0.7838	0.5315	25.0726	0.7706	0.4815	23.8123	0.7285	0.4159
	PSO	25.8470	0.7918	0.5338	25.2894	0.7745	0.4991	24.0461	0.7357	0.4226
	DE	25.7867	0.7892	0.5319	25.2267	0.7710	0.4883	24.0049	0.7341	0.4220
4	AMFRO	**28.2866**	**0.8493**	**0.7060**	28.0211	0.8397	0.6655	26.6743	0.8063	0.5729
	MFRO	27.9708	0.8455	0.6860	27.8096	0.8395	0.6395	26.6743	0.806	0.5719
	EO	27.8384	0.8416	0.6339	27.7522	0.8357	0.6227	26.5527	0.8014	0.5527
	HHO	27.9156	0.8431	0.6513	27.7579	0.8380	0.6259	26.6362	0.8031	0.5556
	SFO	27.8384	0.8395	0.6336	27.5132	0.8300	0.6192	26.5225	0.8010	0.5491
	GWO	27.5013	0.8384	0.5982	27.3039	0.8271	0.5802	26.4156	0.7954	0.5372
	PSO	27.6104	0.8385	0.6243	27.3867	0.8298	0.6043	26.5181	0.8006	0.5469
	DE	27.5702	0.8385	0.6003	27.3485	0.8289	0.5802	26.5085	0.8002	0.5468
5	AMFRO	**29.8792**	**0.8858**	**0.7960**	29.6114	0.8809	0.7426	28.4174	0.8483	0.6702
	MFRO	29.4214	0.8810	0.7409	29.3456	0.8737	0.7402	28.2790	0.8475	0.6512
	EO	29.3257	0.8764	0.7393	29.2131	0.8714	0.7101	28.1821	0.8458	0.6507
	HHO	29.4195	0.8799	0.7394	29.2525	0.8720	0.7183	28.1980	0.8460	0.6510
	SFO	29.2305	0.8739	0.7292	29.2034	0.8702	0.6887	28.1642	0.8455	0.6473
	GWO	28.8094	0.8722	0.6513	28.6577	0.8618	0.6498	27.9538	0.8395	0.6302
	PSO	29.2305	0.8739	0.7207	29.0933	0.8699	0.6809	28.0727	0.8454	0.6395
	DE	29.2026	0.8732	0.7034	28.9685	0.8659	0.6663	27.9538	0.8395	0.6331

Table 7
Comparison of optimal threshold values obtained from different algorithms for the test image with identification number 102 from AANlIB MR image datasets.

K	Methods	Algorithms							
		AMRFO	MRFO	EO	HHO	SFO	GWO	PSO	DE
2	3D Tsallis	72 112	74 121	74 122	74 120	78 128	78 139	77 139	74 121
	2D Tsallis	77 139	78 139	89 164	78 139	77 137	80 129	79 128	89 164
	1D Tsallis	88 164	88 166	91 144	89 165	88 164	88 165	89 164	91 144
3	3D Tsallis	48 96 139	55 101 140	57 101 139	58 106 136	62 109 140	63 105 139	59 101 141	70 102 144
	2D Tsallis	57 106 157	60 109 157	65 112 156	60 110 157	64 119 161	64 115 162	60 109 156	65 112 156
	1D Tsallis	66 126 184	67 125 185	67 126 185	67 126 184	67 127 184	66 126 185	66 126 184	67 126 185
4	3D Tsallis	38 70 105 146	43 80 121 162	52 95 124 153	45 81 116 158	50 93 130 168	53 93 130 166	44 82 123 163	49 88 122 165
	2D Tsallis	41 69 121 164	51 94 132 171	48 97 144 192	52 93 131 166	56 98 127 157	56 90 116 145	51 98 146 195	48 97 144 192
	1D Tsallis	51 98 146 195	51 99 147 195	41 69 121 164	54 105 151 197	50 99 147 194	53 101 147 194	58 91 121 155	51 98 146 195
5	3D Tsallis	28 65 87 110 144	33 72 97 123 156	36 56 94 125 159	38 71 105 139 174	38 64 107 130 158	44 80 115 146 180	39 72 106 143 180	37 74 101 142 190
	2D Tsallis	34 67 104 137 175	47 76 97 121 151	41 65 97 120 148	27 53 78 103 142	48 85 118 145 174	42 85 125 164 204	43 84 124 165 205	41 65 97 120 148
	1D Tsallis	34 68 104 142 193	42 83 124 164 204	45 86 133 175 212	34 68 104 142 193	45 89 131 168 206	53 91 121 149 182	53 87 112 138 167	45 86 133 175 212

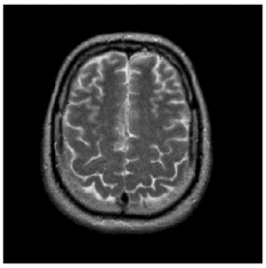

Fig. 11. Test image with identification number 102 from the AANLIB dataset.

The role of the AMRFO algorithm is to maximize the 3D Tsallis entropy efficiently for obtaining these optimal thresholds. The search for optimal threshold vector started with generating the initial Manta Ray population of the AMRFO algorithm randomly. Each Manta Ray (X_i) in the population denotes a possible solution to a threshold vectors set (s_i, t_i, r_i). The dimension K of each vector is decided upon by the number of thresholds required to perform multilevel thresholding. During the progress of the algorithm, each Manta Ray updates their position using the position update rule discussed in Section 3. During every iteration, the fitness value associated with a new position of each Manta Ray compared with the fitness of the best position X_{best} obtained so far and update the X_{best} if possess higher fitness value. Once the stopping criteria met, the optimal threshold vectors (s^*, t^*, r^*) corresponds to the X_{best} is retrieved and the optimal threshold vector contains K threshold values $\left[\frac{s_1^*+t_1^*+r_1^*}{3}, \frac{s_2^*+t_2^*+r_2^*}{3} \ldots\ldots \frac{s_K^*+t_K^*+r_K^*}{3}\right]$ is used for producing the required thresholded image.

The following reconstruction rule is adopted for the generation of output thresholded images. For a threshold vector of K dimension $[Th_1, Th_2, \ldots\ldots Th_K]$, pixels having intensity values less than or equal to Th_1 are replaced by the average intensity of all pixels within this range. Similarly, pixels intensity values within the range $[Th_i+1, Th_{i+1}]$ and $[Th_K + 1, L - 1]$ is assigned a value equal to the average pixels intensity values within the respective ranges.

The flow chart of the method is shown in Fig. 10.

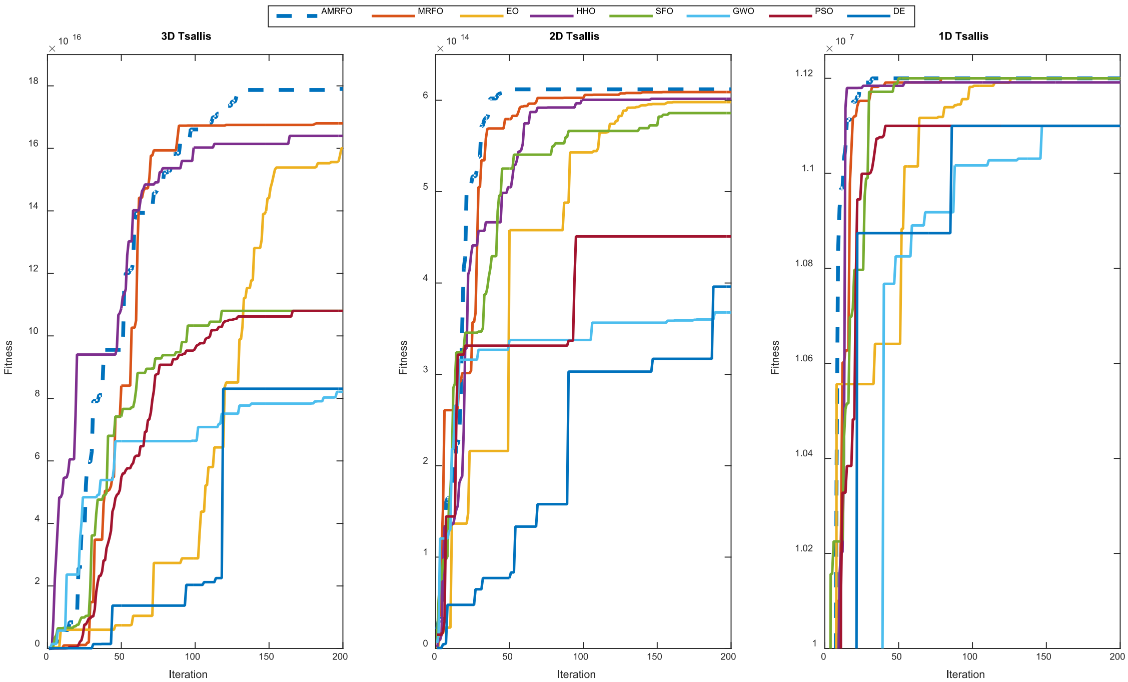

Fig. 12. Convergence plot of various optimization algorithms of the test image with identification number 102 from AANLIB MR image dataset for threshold level $(K) = 5$.

The steps involved in the proposed multilevel thresholding methods are discussed below:

i *Input the image needs to be segmented with a required number of thresholds as dimension K of AMRFO.*
ii *Set the parameters of AMRFO.*
iii *Construct the 3D histogram of the image using mean, median and pixel intensity.*
iv *Input the 3D histogram data and threshold level into the AMRFO algorithm for calculation of entropy as the objective function.*
v *Initialized the position of N number of Manta Ray of dimension D.*
vi *Run the AMRFO algorithm to maximize the objective function till the termination criterion does not arrive.*
vii *Output the desired optimal threshold values.*
viii *Reconstruct the thresholded image using obtained threshold values.*

Table A.1
Unimodal test function.

Function	D	Range	f_{min}
$f_1(X) = \sum_{i=1}^{d} x_i^2$	30	$[-100, 100]^D$	0
$f_2(X) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	30	$[-10, 10]^D$	0
$f_3(X) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100, 100]^D$	0
$f_4(X) = \max_i \left\{ \lvert x_i \rvert, 1 \le i \le d \right\}$	30	$[-100, 100]^D$	0
$f_5(X) = \sum_{i=1}^{d-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	30	$[-30, 30]^D$	0
$f_6(X) = \sum_{i=1}^{d} \left(\lfloor x_i + 0.5 \rfloor \right)^2$	30	$[-100, 100]^D$	0
$f_7(X) = \sum_{i=1}^{d} i x_i^4 + random[0, 1)$	30	$[-1.28, 1.28]^D$	0

6. Results and discussions

In this paper, the proposed 3D Tsallis entropy along with 2D and 1D Tsallis entropy is considered as the fitness function and is maximized using different optimization algorithms to obtain optimal threshold values. For our experiments, 100 different test images are selected from the AANLIB MR image dataset (The Whole Brain Atlas). To evaluate the efficacy of the proposed AMRFO base multilevel thresholding technique, the experimental results of the proposed method is compared with other well-known methods suitable for thresholding—MRFO (Zhao et al., 2020), EO (Faramarzi et al., 2020), HHO (Wunnava et al., 2020a), SFO (Shadravan et al., 2019), GWO (Khairuzzaman and Chaudhury, 2017), PSO (Maitra and Chatterjee, 2008b) and DE (Sarkar and Das, 2013a). Quantitative Performance measures recommended for evaluation of segmentation results are discussed in Table 4.

During the experiment, the population size, and the number of iterations for all algorithms are fixed to 30 and 200, respectively. The Tsallis entropy index α is used as a tuning parameter and must be chosen carefully in the range $[0.1 - 0.3]$ for better segmentation output. In this paper, it is set to 0.1 for threshold selection by maximizing Tsallis entropy. Proposed 3D Tsallis entropy along with Tsallis entropy obtained from 2D and 1D histograms are used as the fitness function for threshold selection in three different multilevel thresholding methods. To show the performance of various thresholding method, we have used the average fitness value (f_{avg}) and standard deviation (std) computed from 100 test images. As the 100 test images have different modalities and features, we obtained a large variation in their fitness values which makes the standard deviation quite high. For an instant, let us consider a simple example to illustrate it, with the help of results obtained from 5 test images having identification numbers 022, 032, 092, 102, and 112. The fitness values associated with these images are 2.25E+04, 2.45E+04, 2.14E+04, 2.04E+04 and 1.87E+04 respectively for a threshold level K=2. This gives an average fitness value of 2.15E+04 and a standard deviation of 2.18E+03. It can be observed that the fitness value of one image is quite different from the other, which leads to a large standard deviation. Similarly, the average fitness value (f_{avg}) and standard deviation (std) of 100 test images for threshold levels $K = 2, 3, 4$, and 5, which are evaluated using 21 independent runs of each image are presented in Table 5.

A higher fitness value with a lower standard deviation of a thresholding method indicates the better one. It is clear from the 1D Tsallis

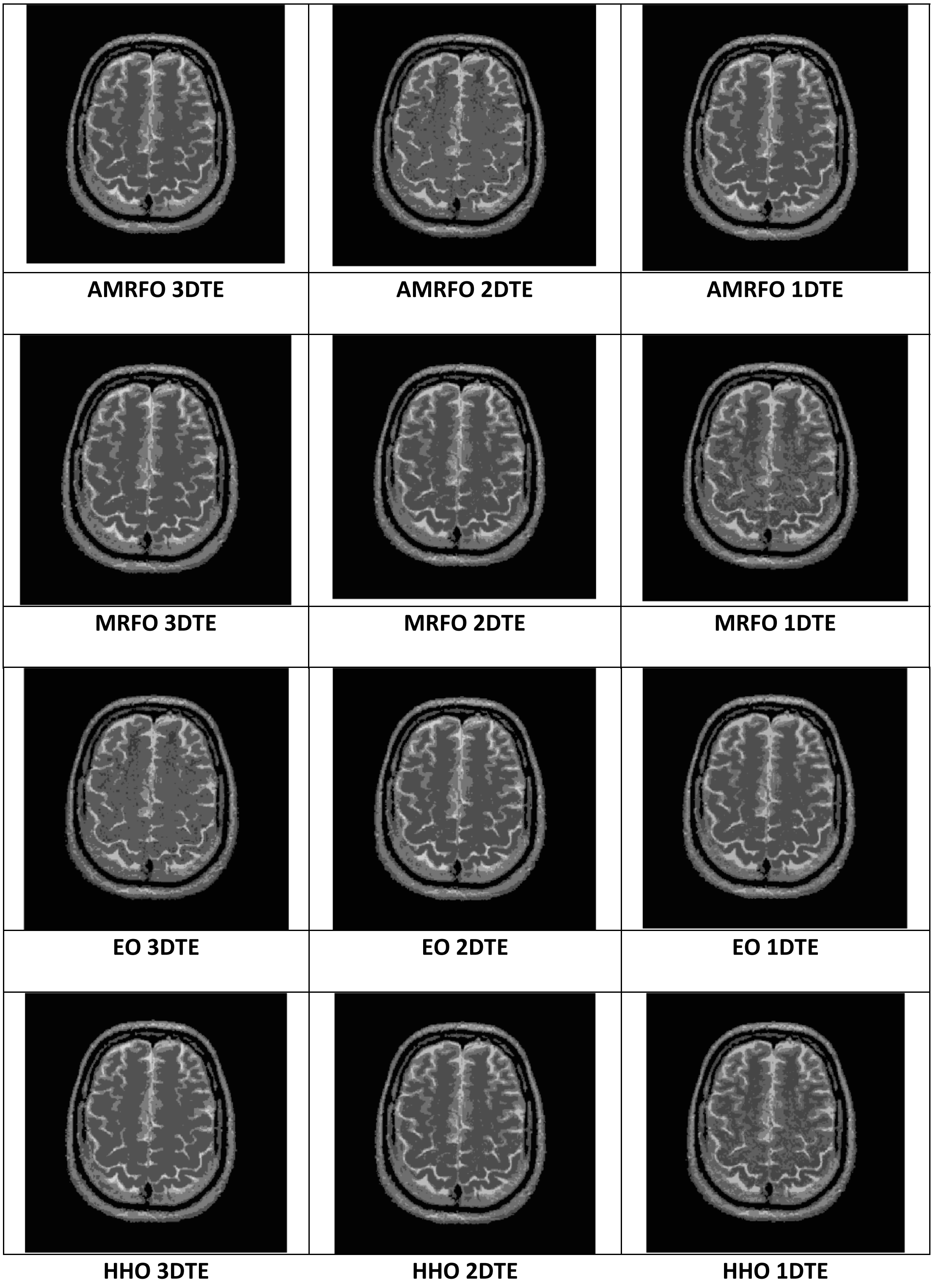

Fig. 13. Visual Comparison of Thresholded images obtained from various methods of the test image with identification number 102 from AANLIB MR image dataset for threshold level $(K) = 5$.

entropy result, the performance of all search algorithms is almost similar for the average fitness values. Whereas there is a deviation in the performance of search algorithms when moving from 1D to 3D Tsallis based Thresholding. There are significant changes in fitness function for 3D Tsallis for various optimization algorithms at a higher level of thresholding due to more complexity in threshold selection from a 3D histogram. Higher fitness value and lower standard deviation for the different threshold level of AMRFO algorithms shows its superiority over compared optimization algorithms. To show the effectiveness of the 3D Tsallis based technique in multilevel thresholding, average PSNR, FSIM, and SSIM value obtained from different methods presented in Table 6. Higher the PSNR, FSIM, and SSIM values indicate better reconstruction of the segmented image from obtained threshold values. It is clear from the table that, there is a substantial improvement of PSNR values for 3D Tsallis entropy-based thresholding at a higher level of thresholding. Two more similarity measures FSIM

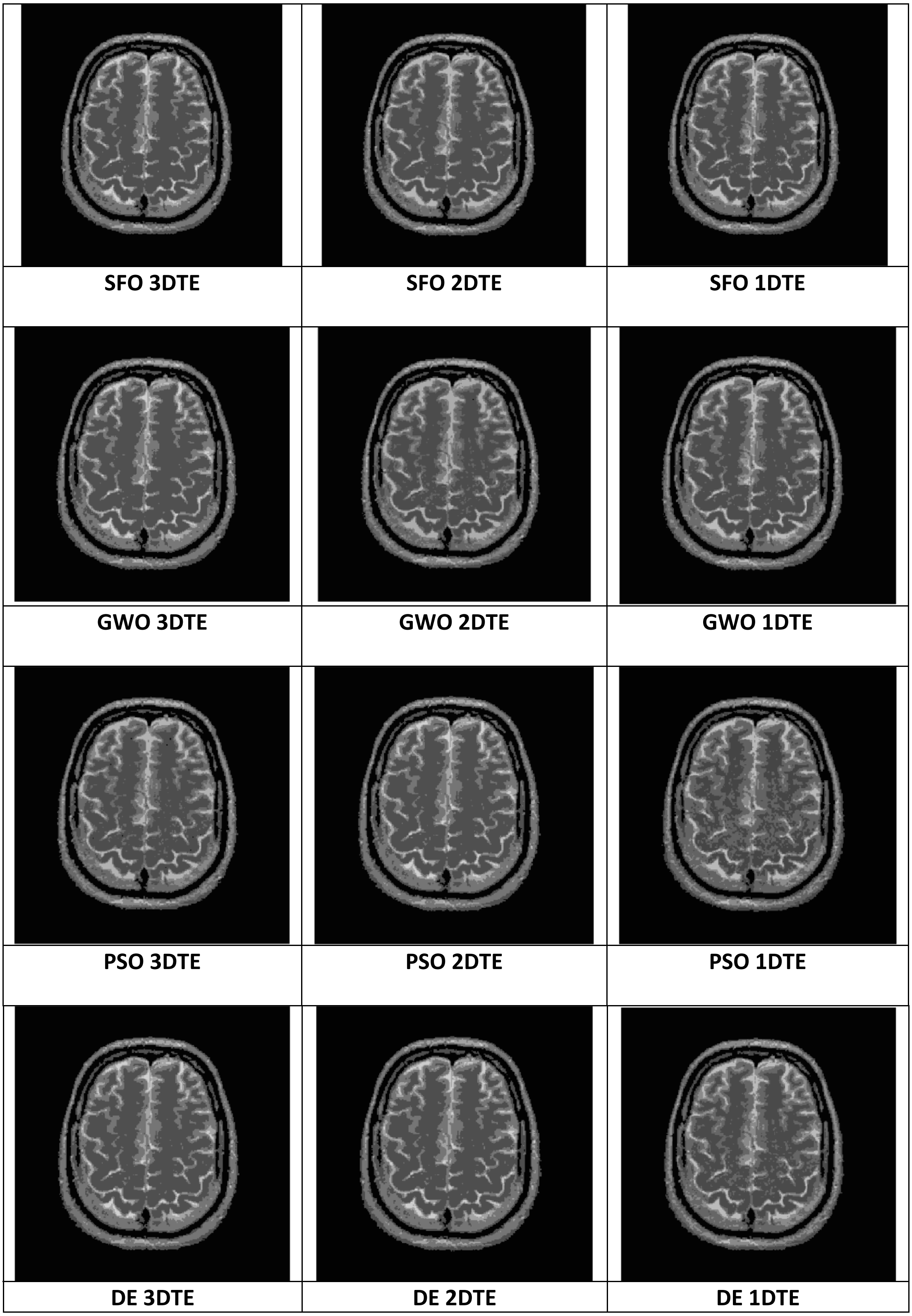

Fig. 13. (*continued*).

and SSIM also available in the table to support the effectiveness of the proposed method of thresholding. Higher values of FSIM and SSIM values indicating better segmentation results are marked with bold font in the table.

The result analysis is further extended to explore the convergence rate and visual assessment of the different methods on a test image with identification number 102 from the dataset, which is shown in Fig. 11. The convergence curves of AMRFO and other optimization algorithms for threshold level 5 using 3D, 2D, and 1D Tsallis entropy as fitness function are shown in Fig. 12. It can be observed that for 1D Tsallis entropy AMRFO, MRFO, HHO, and SFO are performing well. For 2D and 3D Tsallis entropy function, the convergence curve of AMRFO

Table A.2
Scalable dimension multimodal test function.

Function	D	Range	f_{min}
$f_8(X) = \sum_{i=1}^{d} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	30	$[-5.12, 5.12]^D$	0
$f_9(X) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos\left(2\pi x_i\right)\right) + 20 + e$	30	$[-32, 32]^D$	0
$f_{10}(X) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	0
$f_{11}(X) = \frac{\pi}{d}\left\{10\sin\left(\pi y_i\right) + \sum_{i=1}^{d-1}\left(y_i - 1\right)^2\left[1 + 10\sin 2\left(\pi y_{i+1}\right)\right] + \left(y_d - 1\right)^2\right\} + \sum_{i=1}^{d} u\left(x_i, 10, 100, 4\right)$			
$y_i = 1 + \frac{x_i + 1}{4} u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m & x_i > a \\ 0 & -a < x_i < a \\ k\left(-x_i - a\right)^m & x_i < -a \end{cases}$	30	$[-50, 50]^D$	0
$f_{12}(X) = 0.1\left\{\sin 2\left(3\pi x_1\right) + \sum_{i=1}^{d}\left(x_i - 1\right)^2\left[1 + \sin 2\left(3\pi x_i + 1\right)\right] + \left(x_d - 1\right)^2\left[1 + \sin 2\left(2\pi x_d\right)\right]\right\} + \sum_{i=1}^{d} u\left(x_i, 5, 100, 4\right)$			
$y_i = 1 + \frac{x_i + 1}{4} u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m & x_i > a \\ 0 & -a < x_i < a \\ k\left(-x_i - a\right)^m & x_i < -a \end{cases}$	30	$[-50, 50]^D$	0

Table A.3
Fixed dimension multimodal test function.

Function	Range	f_{min}
$f_{13}(X) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5, 5]^4$	0.0003075
$f_{14}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	−1.0316285
$f_{15}(X) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$[-5, 10] \times [0, 15]$	0.398
$f_{16}(X) = \left[1 + \left(x_1 + x_2 + 1\right)^2 \times \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2\right)\right] \times$ $\left[30 + \left(2x_1 - 3x_2\right)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2\right)\right]$	$[-2, 2]^2$	3
$f_{17}(X) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	$[0, 1]^3$	−3.86
$f_{18}(X) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	$[0, 1]^6$	−3.32
$f_{19}(X) = -\sum_{i=1}^{5}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.1532
$f_{20}(X) = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.4028
$f_{21}(X) = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.5363

dominates all by attending the maximum fitness value in a short time. The corresponding optimal threshold values are presented in Table 7. It is observed that there is a significant difference in the threshold value of AMRFO from others in the 3D Tsallis entropy method of thresholding. Quantitative analysis is not always preferred to judge the quality of the segmentation. Thus, sometimes visual assessment of the segmentation outputs is also considered for quality judgment. Therefore, we have generated the thresholded images of 100 test images at a different level of thresholding and investigated them visually. The results of one of the test images with identification number 102 of the AANLIB dataset are shown in Fig. 13. It can be observed from the output images of different methods that the proposed 3D Tsallis based thresholding output is optimum in quality than others.

Table A.4
Composition test function from CEC 2014.

Function	Name of the function	Range	f_{min}
$f_{22}(X)$	Composition Function 1 ($N = 5$)	$[-100, 100]^D$	2300
$f_{23}(X)$	Composition Function 2 ($N = 3$)	$[-100, 100]^D$	2400
$f_{24}(X)$	Composition Function 3 ($N = 3$)	$[-100, 100]^D$	2500
$f_{25}(X)$	Composition Function 4 ($N = 5$)	$[-100, 100]^D$	2600
$f_{26}(X)$	Composition Function 5 ($N = 5$)	$[-100, 100]^D$	2700
$f_{27}(X)$	Composition Function 6 ($N = 5$)	$[-100, 100]^D$	2800

7. Conclusion

A 3D Tsallis entropy-based multilevel thresholding method using a new search algorithm called attacking Manta Ray foraging optimization (AMRFO) is presented. The AMRFO algorithm is found well suited for optimization, because of its inherent attacking power. This additional power makes the Manta Ray more attacking in the exploration stage to avoid the local minima in its path and move towards the optimal solution. The convergence rate of AMRFO is also quite impressive than other state-of-the-art algorithms. The performance of AMRFO is explicitly shown using benchmark functions. The 3D histogram is constructed with an average and median value of the neighboring pixel intensity with the original pixel point. Therefore, the information related to the spatial distribution of pixel intensity and boundary points is retained. This helps the 3D histogram-based entropy calculation more appropriate for optimal threshold selection. Nonetheless, the use of the median makes it more attractive for such applications. The proposed 3D Tsallis entropy concept may enrich the literature. The statistical comparison using average fitness functions, PSNR, FSIM, and SSIM presented in the paper claims the superiority of the proposed method over others. Wilcoxon's signed-rank test and Friedman's mean rank test results exhibit its enforcement in the subject field. For qualitative analysis, the segmented output images at different levels of thresholding are also shown. The proposed method yields better quality outputs because the 3D construction of the histogram retains more edge information. The work may encourage researchers to explore its capability for solving the world of engineering problems. Its future application would be the segmentation of the multi-spectral color images.

CRediT authorship contribution statement

Bibekananda Jena: Conceptualization, Implementation, Data handling, Programming, Writing - original draft. **Manoj Kumar Naik:** Methodology, Guidance, Investigation, Revision of draft. **Rutuparna Panda:** Methodology, Guidance, Data analysis, Revision of draft. **Ajith Abraham:** Supervision, Analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

See Tables A.1–A.4.

References

Abdel-Basset, M., Chang, V., Mohamed, R., 2020. A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. Neural Comput. Appl. http://dx.doi.org/10.1007/s00521-020-04820-y.

Agrawal, S., Panda, R., Bhuyan, S., Panigrahi, B.K., 2013. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol. Comput. 11, 16–30. http://dx.doi.org/10.1016/j.swevo.2013.02.001.

Alihodzic, A., Tuba, M., 2014. Improved bat algorithm applied to multilevel image thresholding. J. Sci. World J. 2014 (16), http://dx.doi.org/10.1155/2014/176718.

Alwerfali, H.S.N., Abd Elaziz, M., Al-Qaness, M.A.A., Abbasi, A.A., Lu, S., Liu, F., Li, L., 2019. A multilevel image thresholding based on hybrid salp swarm algorithm and fuzzy entropy. IEEE Access 7, 181405–181422. http://dx.doi.org/10.1109/ACCESS.2019.2959325.

Baby Resma, K.P., Nair, M.S., 2018. Multilevel thresholding for image segmentation using krill herd optimization algorithm. J. King Saud Univ. - Comput. Inform. Sci. http://dx.doi.org/10.1016/j.jksuci.2018.04.007.

Bhandari, A., Kumar, I., Srinivas, K., 2019. Cuttlefish algorithm based multilevel 3D otsu function for color image segmentation. IEEE Trans. Instrum. Meas. 1. http://dx.doi.org/10.1109/TIM.2019.2922516.

Bhandari, A.K., Singh, V.K., Kumar, A., Singh, G.K., 2014. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst. Appl. 41, 3538–3560. http://dx.doi.org/10.1016/j.eswa.2013.10.059.

Bohat, V.K., Arya, K.V., 2019. A new heuristic for multilevel thresholding of images. Expert Syst. Appl. 117, 176–203. http://dx.doi.org/10.1016/j.eswa.2018.08.045.

Chao, Y., Dai, M., Chen, K., Chen, P., Zhang, Z., 2016. Fuzzy entropy based multilevel image thresholding using modified gravitational search algorithm. In: Proceedings of the IEEE International Conference on Industrial Technology 2016-May. pp. 752–757. http://dx.doi.org/10.1109/ICIT.2016.7474845.

Daoud, A.O., Tsehayae, A.A., Fayek, A.R., 2017. A guided evaluation of the impact of research and development partnerships on university, industry, and government. Can. J. Civil Eng. 44, 253–263. http://dx.doi.org/10.1139/cjce-2016-0381.

Education, H., 2019. A multilevel image thresholding approach based on crow search algorithm and Otsu method. J. Decis. Operat. Res. http://dx.doi.org/10.22105/DMOR.2019.88580.

Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S., 2020. Equilibrium optimizer: A novel optimization algorithm. Knowl.-Based Syst. 191, 105190. http://dx.doi.org/10.1016/j.knosys.2019.105190.

He, L., Huang, S., 2017. Modified firefly algorithm based multilevel thresholding for color image segmentation. Neurocomputing 240, 152–174. http://dx.doi.org/10.1016/j.neucom.2017.02.040.

Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H., 2019. Harris hawks optimization: Algorithm and applications. Future Gener. Comput. Syst. 97, 849–872. http://dx.doi.org/10.1016/j.future.2019.02.028.

Horng, M.H., 2010. Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst. Appl. 37, 4580–4592. http://dx.doi.org/10.1016/j.eswa.2009.12.050.

Horng, M.H.M.-H., 2011. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst. Appl. 38, 13785–13791. http://dx.doi.org/10.1016/j.eswa.2011.04.180.

Horng, M.H.M.-H., Liou, R.J.R.-J., 2011. Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Syst. Appl. 38, 14805–14811. http://dx.doi.org/10.1016/j.eswa.2011.05.069.

Jain, M., Singh, V., Rani, A., 2019. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol. Comput. 44, 148–175. http://dx.doi.org/10.1016/j.swevo.2018.02.013.

Kapur, J.N., Sahoo, P.K., Wong, A.K.C., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. Comput. Vis., Grap., Image Process. 29, 273–285. http://dx.doi.org/10.1016/0734-189X(85)90125-2.

Khairuzzaman, A.K.M., Chaudhury, S., 2017. Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst. Appl. 86, 64–76. http://dx.doi.org/10.1016/j.eswa.2017.04.029.

Liang, H., Jia, H., Xing, Z., Ma, J., Peng, X., 2019. Modified grasshopper algorithm-based multilevel thresholding for color image segmentation. IEEE Access 7, 11258–11295. http://dx.doi.org/10.1109/ACCESS.2019.2891673.

Liang, J.J., Qu, B.Y., Suganthan, P.N., 2013. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization.

Lin Zhang, L.Z., 2011. FSIM: A feature similarity index for image quality assessment. IEEE Trans. Image Process. 20 (2378), http://dx.doi.org/10.1109/TIP.2011.2109730.

Liu, J., Li, W., Tian, Y., 1991. Automatic thresholding of gray-level pictures using two-dimensional Otsu method. pp. 325–327.

Maitra, M., Chatterjee, A., 2008a. A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst. Appl. 34, 1341–1350. http://dx.doi.org/10.1016/j.eswa.2007.01.002.

Maitra, M., Chatterjee, A., 2008b. A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst. Appl. 34, 1341–1350. http://dx.doi.org/10.1016/j.eswa.2007.01.002.

Masi, M., 2005. A step beyond Tsallis and Rényi entropies. Phys. Lett., A: General, Atomic Solid State Phys. 338, 217–224. http://dx.doi.org/10.1016/j.physleta.2005.01.094.

Mlakar, U., Potočnik, B., Brest, J., 2016. A hybrid differential evolution for optimal multilevel image thresholding. Expert Syst. Appl. 65, 221–232. http://dx.doi.org/10.1016/j.eswa.2016.08.046.

Nie, F., Wang, Y., Pan, M., Peng, G., Zhang, P., 2013. Two-dimensional extension of variance-based thresholding for image segmentation. Multidimens. Syst. Signal Process. 24, 485–501. http://dx.doi.org/10.1007/s11045-012-0174-7.

Otsu, 1979. Otsu_1979_otsu_method. IEEE Trans. Syst., Man, Cybern. C 6, 2–66. http://dx.doi.org/10.1109/TSMC.1979.4310076.

Sahoo, P., Wilkins, C., Yeager, J., 1997. Threshold selection using Renyi's entropy. Pattern Recognit. 30, 71–84. http://dx.doi.org/10.1016/S0031-3203(96)00065-9.

Saremi, S., Mirjalili, S., Lewis, A., 2017. Grasshopper optimisation algorithm: Theory and application. Adv. Eng. Softw. 105, 30–47. http://dx.doi.org/10.1016/j.advengsoft.2017.01.004.

Sarkar, S., Das, S., 2013a. Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy— A differential evolution approach. IEEE Trans. Image Process. 22, 4788–4797. http://dx.doi.org/10.1109/TIP.2013.2277832.

Sarkar, S., Das, S., 2013b. Multilevel image thresholding based on 2D histogram and maximum tsallis entropy - A differential evolution approach. IEEE Trans. Image Process. 22, 4788–4797. http://dx.doi.org/10.1109/TIP.2013.2277832.

Shadravan, S., Naji, H.R., Bardsiri, V.K., 2019. The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng. Appl. Artif. Intell. 80, 20–34. http://dx.doi.org/10.1016/j.engappai.2019.01.001.

Shubham, S., Bhandari, A.K., 2019. A generalized masi entropy based efficient multilevel thresholding method for color image segmentation. Multimedia Tools Appl. 78, 17197–17238. http://dx.doi.org/10.1007/s11042-018-7034-x.

Sun, M., Wei, H., 2020. An improved cuckoo search algorithm for multi-level gray-scale image thresholding. Multimedia Tools Appl. http://dx.doi.org/10.1007/s11042-020-08931-5.

The Whole Brain Atlas, 2015.

Wang, Q., Zhao, H., Wu, W., Yuan, N., 2012. Algorithm for segmentation based on an improved three-dimensional Otsu's thresholding. In: Proceedings of 2nd International Conference on Computer Science and Network Technology, ICCSNT 2012. pp. 1737–1740. http://dx.doi.org/10.1109/ICCSNT.2012.6526256.

Williamson, D.F., Parker, A, R., Kendrick, S, J., 1989. The box plot: a simple visual method to interpret data. Ann. Internal Med. 110, 916–921.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82. http://dx.doi.org/10.1109/4235.585893.

Wunnava, A., Kumar Naik, M., Panda, R., Jena, B., Abraham, A., 2020a. A differential evolutionary adaptive harris hawks optimization for two dimensional practical masi entropy-based multilevel image thresholding. J. King Saud Univ. - Comput. Inform. Sci. http://dx.doi.org/10.1016/j.jksuci.2020.05.001.

Wunnava, A., Naik, M.K., Panda, R., Jena, B., Abraham, A., 2020b. An adaptive Harris hawks optimization technique for two dimensional grey gradient based multilevel image thresholding. Appl. Soft Comput. 106526. http://dx.doi.org/10.1016/j.asoc.2020.106526.

Zaitoun, Nida M., Aqel, M.J., 2015. Survey on image segmentation techniques. Procedia Comput. Sci. 65, 797–806. http://dx.doi.org/10.1016/j.procs.2015.09.027.

Zhao, W., Zhang, Z., Wang, L., 2020. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Eng. Appl. Artif. Intell. 87, 103300. http://dx.doi.org/10.1016/j.engappai.2019.103300.

Zhiwei, Y., Zhaobao, Z., Xin, Y., Xiaogang, N., 2005. Automatic threshold selection based on ant colony optimization algorithm. In: 2005 International Conference on Neural Networks and Brain. pp. 728–732. http://dx.doi.org/10.1109/ICNNB.2005.1614730.