Engineering Applications of Artificial Intelligence 94 (2020) 103836

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A novel interdependence based multilevel thresholding technique using adaptive equilibrium optimizer

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

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

ARTICLE INFO

Keywords:
Artificial intelligence
Interdependency
Entropic function
Soft computing
Optimization techniques
Multilevel image thresholding

ABSTRACT

The multilevel thresholding is one important operation in computer vision, which is a subfield of artificial intelligence (AI), used to understand and interpret the data in the real world. The existing entropic methods, based on the histogram of an image for the multilevel thresholding mostly deal with the maximization of the entropic information excluding the shred boundary, which reduces the accuracy. These problems lead to the poor thresholding accuracy and a lower speed. To address the problem, we propose a novel interdependence based technique that uses the shred boundary, which is a minimization problem. A firsthand objective function is investigated, which takes care of the shred boundary. The traditional multilevel thresholding techniques are computationally expensive due to the exhaustive search process, an alternate method is to use the evolutionary computation based on a nature-inspired algorithm. In this paper, a new optimizer called adaptive equilibrium optimizer (AEO) is also proposed for multilevel thresholding, an improvement over the basic equilibrium optimizer (EO) by implementing an adaptive decision making of dispersal for nonperformer search agents. The AEO performance is compared with state-of-the-art algorithms — equilibrium optimizer (EO), gray wolf optimizer (GWO), whale optimization algorithm (WOA), squirrel search algorithm (SSA) and the wind driven optimization (WDO) algorithm, using standard benchmark functions. Based on the qualitative and quantitative analysis, the AEO outperformed EO, GWO, WOA, SSA, and WDO. The optimal thresholds are obtained by minimizing the objective function using the AEO. For the experiment, 500 images of the BSDS 500 dataset are considered. Popular metrics such as the peak signal to noise ratio (PSNR), structural similarity index (SSIM), and the feature similarity index (FSIM) are considered for a quantitative analysis. Remarkable differences in the thresholding accuracy are observed with a simultaneously decreasing computational complexity. The merits of the paper are highlighted to ensure its future use in the world of engineering applications using soft computing, a subfield of the AI.

1. Introduction

Image segmentation is a fundamental step of the image processing in which an image is segmented into different segments based on its characteristics — the intensity, contrast, and texture. Image thresholding is the simplest approach in the image segmentation. It uses the basic assumption of the object and the background that has distinct gray level distributions. Based on this assumption, the gray level histogram contains multiple picks. The corresponding threshold values can be obtained for the multilevel image thresholding. The various global thresholding methods are described in Fu and Mui (1981), Pal and Pal (1993), Sahoo et al. (1988), Sankur and Sezgin (2001), Sezgin and Sankur (2004) and Zhang et al. (2008). The thresholding methods are classified by parametric and non-parametric approaches. The non-parametric approach of thresholding methods uses some maximization criteria to obtain the threshold values (Bohat and Arya, 2019). Some of the non-parametric classical thresholding methods are Otsu's method (Otsu, 1979), Kapur's entropy (Kapur et al., 1985), Tsallis entropy (Pavesic and Ribaric, 2000; Portes de Albuquerque et al., 2004; Tsallis, 2001) and Masi entropy (Khairuzzaman and Chaudhury, 2019; Masi, 2005; Nie et al., 2017) based on maximization criterion determined

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

https://doi.org/10.1016/j.engappai.2020.103836
Received 18 March 2020; Received in revised form 6 June 2020; Accepted 15 July 2020
Available online 28 July 2020

from the histogram of a target image. Otsu's thresholding method is a nonparametric and unsupervised nature of the threshold selection based on discriminant criterion by maximizing the intra-class variance among the foreground and background classes. The Kapur's entropy obtained the threshold by using the maximization criterion of the entropic information among the different classes. The Tsallis entropy is a non-extensive entropy, which is an extension to Boltzmann–Gibbs entropy. Nature-inspired algorithms are a more practical approach to lower the computational complexity of multilevel thresholding. The Masi entropy is a generalized entropy that can manage the additive/nonextensive information that occurs in a physical system. Apart from these techniques, another thresholding method named Johannsen and Bille method (Johannsen and Bille, 1982) use entropic information from the histogram for bi-level thresholding by minimizing the interdependence of two classes and show some impressive results as described in Sahoo et al. (1988).

The thresholding methods are classified as the bi-level thresholding and the multilevel thresholding based on the number of thresholds used to segment the image. The bi-level threshold uses one threshold value to segment the entire image into the foreground and the background regions. However, a set of two or more thresholds is used to segment the whole image in the multilevel thresholding. The multilevel thresholding is strongly recommended over bi-level thresholding for superior performances (Agrawal et al., 2017). As the number of thresholds increases in the multilevel thresholding, it provides superior performance with a cost of computational time complexity. The computational time increases due to the complexity of a simple exhaustive search, which is $O\left(L^K\right)$ for K thresholds (Yin, 1999). The complexity can be reduced using recursive algorithms (Liao et al., 2001; Peng-Yeng and Ling-Hwei, 1994; Yin and Chen, 1997) to obtain the optimal thresholds with the help of lookup tables. When the number of thresholds increases, the computation time increases even by using the recursive algorithms (Song et al., 2017). Therefore, the problem still exists. To overcome the computation time complexity, nature-inspired optimization algorithms are utilized in the multilevel thresholding.

The recent developments in the nature-inspired optimization algorithms have shown their strong potential to be used in the optimization problems. These algorithms are — equilibrium optimizer (EO) (Faramarzi et al., 2020) inspired by dynamic and equilibrium state of control volume mass balance model, gray wolf optimizer (GWO) (Mirjalili et al., 2014) stimulated by leadership and hunting mechanism of gray wolves, whale optimization algorithm (WOA) (Mirjalili and Lewis, 2016) inspired by social behavior of humpback whales, squirrel search algorithm (SSA) (Jain et al., 2019) motivated by foraging behavior of southern flying squirrels, and wind driven optimization (WDO) (Bayraktar et al., 2013) inspired by air parcels in N-dimensional space using Newton's second law is good to obtain the optimal solution on its capacity. Sometimes a basic version of the algorithm is not perfect for all types of problems, so modifying or hybridizing the algorithms improves the performance. In this regard, some recent advancements in the optimization are — GWO with fuzzy weight hierarchical operator (Rodríguez et al., 2017), continuous Karnik–Mendel method using the Type-2 fuzzy system for harmony search (HS) and differential evolution (DE) (Castillo et al., 2019), fuzzy adaptive multi-objective evolutionary algorithm (FAME) (Santiago et al., 2019) based on fuzzy interference system, multi-objective hierarchical genetic algorithm (MOHGA) (Melin and Sánchez, 2018) optimization of modular neural network and construed real parameter optimization (Rodríguez et al., 2020) using a firefly algorithm (FA) and gray wolf optimizer (GWO).

To overcome the computational time complexity issue in the multilevel thresholding, researchers extensively use soft computing, a subfield of the AI. The gray wolf optimizer (GWO) is used to obtain the threshold value in the multilevel thresholding using Otsu's between-class variance and Kapur's entropy in Khairuzzaman and Chaudhury (2017), which shows better performance than the particle swarm optimization (PSO) and bacteria foraging optimization (BFO). A krill herd optimization (KHO) is used to determine the optimal threshold values based on Otsu's and Kapur's entropy as objective functions (Baby Resma and Nair, 2018), which shows that the KHO has an advantage in computational time over BFO, PSO, genetic algorithm (GA) and moth-flame Optimization (MFO). Similarly, El Aziz et al. show better performance in the multilevel thresholding using Otsu's and Kapur's entropy methods with the help of a whale optimization algorithm (WOA) by lowering the processing time (El Aziz et al., 2018). The PSO with Masi entropy as an objective function for multilevel thresholding is proposed in Khairuzzaman and Chaudhury (2019), which shows improved performance over Kapur's entropy. Martino and Sessa propose multilevel thresholding using maximizing the fuzzy entropy with the help of a variant of the PSO named as chaotic Darwinian particle swarm optimization (CDPSO) as an optimizer (Martino and Sessa, 2020), which shows improvements over the results and CPU computational time. The crow search algorithm (CSA) with Kapur's entropy is used to get the optimal threshold values for the multilevel thresholding in Upadhyay and Chhabra (2019), which shows that the CSA present better quality and consistency. Kandhway and Bhandari proposed Masi entropy as an objective function for multilevel thresholding using a water cycle algorithm (WCA) in Kandhway and Bhandari (2019), which interpreted that Masi entropy method outperformed Tsallis entropy. The symbiotic organisms search (SOS) is used to find the optimal threshold values in Küçükuğurlu and Gedikli (2020) with the help of Kapur's entropy, which shows a performance improvement over the PSO, GA, GWO, artificial bee colony (ABC) and the firefly algorithm (FA) approach. Developments of hybridizing/modifying the nature-inspired optimization algorithms, to achieve the best in multilevel thresholding, are found in the literature. The hybrid bat algorithm with genetic crossover operation and smart inertia weight (SGA-BA) (Yue and Zhang, 2020), Hybrid-heuristic method (as a combination of genetic algorithm (GA), sine cosine algorithm (SCA), artificial bee colony (ABC) algorithm, firefly algorithm (FA) and social-spider optimization (SSO) algorithm) (Elaziz et al., 2020) uses the Otsu's and Kapur's entropy for the multilevel thresholding. Recently, differential evolutionary adaptive Harris hawks optimization (DEAHHO) using a novel 2-D practical Masi entropy as an objective function of the multilevel thresholding technique is proposed in Wunnava et al. (2020), which shows the potential of the hybridization of the optimization algorithms to overcome the disadvantage of the basic algorithm. This has motivated the authors to explore a new method for the multilevel image thresholding to enrich the computer vision field of the AI. Although there are several methods (discussed above), this work is an attempt to achieve significant progress in the world of engineering applications, specifically segmentation, in the field computer vision, a subfield of the AI, to understand and interpret the object classification, identification, localization, and description.

The focus of the work is to investigate a novel interdependence based multilevel thresholding technique and an efficient nature-inspired optimization algorithm to achieve optimal threshold values to preserve more information. The idea of the interdependence based bi-level thresholding is found in Johannsen and Bille (1982). This idea is used here to develop new theoretical contributions to the multilevel thresholding of gray level images by minimizing the interdependency. In this context, a novel objective function is investigated to minimize the entropic interdependencies between various classes. This is an interesting contribution of the present work. As the computational time increases with the number of thresholds, an exhaustive search using the traditional procedures is a difficult task, that requires an optimizer. This warrants us to investigate a novel nature-inspired optimization algorithm. For this purpose, recently the equilibrium optimizer (EO) proposed in Faramarzi et al. (2020) is considered. The EO is inspired by the dynamic and the equilibrium states of the control volume mass balance model. In this algorithm, the search agents update the position concerning the equilibrium candidates, to reach the optimal solution (equilibrium state). However,

the search agents position update always depends on the equilibrium state, which may restrict the exploration capability of the EO. To avoid this, we suggest an adaptive position update strategy for the search agents based on the current fitness value; and coined as an adaptive equilibrium optimizer (AEO). This is another interesting contribution of the paper. A qualitative and quantitative analysis of AEO is performed with the help of a set of twenty-three well-known benchmark functions (Liang and Suganthan, 2005; Naik and Panda, 2016; Yao et al., 1999). The performance is compared with state-of-the-art nature-inspired algorithms such as EO, GWO, WOA, SSA, and WDO. The AEO is ranked one based on Friedman mean rank. The AEO performed well, it encourages to employ the AEO as an optimizer to find the optimal threshold values using interdependence based multilevel thresholding. To demonstrate the feasibility and superiority, the method is evaluated using the Berkeley Segmentation Data Set (BSDS 500). The segmented images using the optimal threshold values obtained are evaluated statistically using a well-known Wilcoxon test. Different performance metrics are also considered for quantitative analysis. Thresholded images are displayed in the result and discussion section to provide a qualitative analysis. Based on the nonparametric statistical test (Wilcoxon test), our technique outperforms other methods in multilevel thresholding.

The organization of the paper is as follows. Section 1 is devoted to a brief introduction. The problem formulation based on interdependence based objective function for multilevel thresholding is presented in Section 2. The development of a new optimization algorithm coined as the AEO is presented in Section 3. The proposed method for multilevel image thresholding is described in Section 4. Results and discussions are highlighted in Section 5. Conclusions are drawn in Section 6.

2. Problem formulation

2.1. Multilevel thresholding

Let us consider an image I of size $M \times N$, with L gray levels in the range $[0, L-1]$, then the probability of each gray level is expressed as

$$p_i = \frac{n_i}{M \times N}, i \in [0, 1, \dots, L-1], \tag{1}$$

where n_i represents the number of pixels with the ith gray level and $\sum_{i=0}^{L-1} p_i = 1,\ p_i \geq 0$.

Let the K threshold values $t_1, t_2, \dots, t_K$ divide the image into $K+1$ classes as M_i where $i = 1, 2, \dots, K+1$ in the multilevel thresholding. The M_1 is the foreground class, $M_{i=2,3,\dots,K}$ are the intermediate classes and M_{K+1} is the background class. Then the different classes in the multilevel thresholding are defined as:

$$\begin{aligned} &[0, t_1 - 1] \in M_1 \\ &[t_1, t_2 - 1] \in M_2 \\ &\quad\vdots \\ &[t_K, L-1] \in M_{K+1} \end{aligned} \tag{2}$$

2.2. The proposed interdependence based objective function for multilevel thresholding

The bi-level thresholding method based on the interdependency is found in Johannsen and Bille (1982). The authors use the entropic information form the histogram to divide the gray level distribution into two parts by minimizing an interdependence function. A comparison of their method with other bi-level thresholding techniques is discussed in Sahoo et al. (1988).

The optimal threshold t^* is chosen by minimizing the interdependence using the relation

$$t^* = \arg \min_{0<t<L-1} \left\{ S_f\left(M_f\right) + S_b\left(M_b\right) \right\}, \tag{3}$$

where entropic information of the foreground class M_f consisting of gray level $[0, t]$ is defined as S_f and the other background class M_b consisting of gray level $[t, L-1]$ is defined as S_b. The $S_f\left(M_f\right)$ and $S_b\left(M_b\right)$ are calculated as

$$S_f\left(M_f\right) = \log_e\left(\sum_{i=0}^{t} p_i\right) - \frac{1}{\sum_{i=0}^{t} p_i}\left[p_t \log_e p_t + \left(\sum_{i=0}^{t-1} p_i\right) \log_e\left(\sum_{i=0}^{t-1} p_i\right)\right], \tag{4}$$

and

$$S_b\left(M_b\right) = \log_e\left(\sum_{i=t}^{L-1} p_i\right) - \frac{1}{\sum_{i=t}^{L-1} p_i}\left[p_t \log_e p_t + \left(\sum_{i=t+1}^{L-1} p_i\right) \times \log_e\left(\sum_{i=t+1}^{L-1} p_i\right)\right]. \tag{5}$$

It is observed that the entropic information calculation of S_f and S_b depends on the threshold value t, which is the shred boundary of the foreground and the background class. Let us extend the concept of the bi-level thresholding to the multilevel thresholding for K number of thresholds. This is one of the main contributions of our paper. The corresponding entropic information for the $K+1$ classes is referred to as S_i where $i = 1, 2, \dots, K+1$. The set of gray levels used for the calculation of the entropic information for various classes is displayed in Table 1.

Table 1
Set of gray level and threshold values as the shred boundary for different classes of the entropic information.

Class	Entropic information	Set of gray level	Threshold value as shred boundary
M_1	S_1	$[0, t_1]$	t_1
M_2	S_2	$[t_1, t_2]$	$t_1,\ t_2$
$M_{i=3,4,\dots,K}$	$S_{i=3,4,\dots,K}$	$[t_{i-1}, t_i]$	$t_{i-1},\ t_i$
M_{K+1}	S_{K+1}	$[t_K, L-1]$	t_K

The entropic information $S_{i=1,2,\dots,K+1}$ of the different classes for the multilevel thresholding is formulated as

$$\begin{aligned} S_1\left(M_1\right) &= \log_e\left(\sum_{i=0}^{t_1} p_i\right) - \frac{1}{\sum_{i=0}^{t_1} p_i}\left[\left(\sum_{i=0}^{t_1-1} p_i\right) \times \log_e\left(\sum_{i=0}^{t_1-1} p_i\right) + p_{t_1} \log_e p_{t_1}\right] \\ S_2\left(M_2\right) &= \log_e\left(\sum_{i=t_1}^{t_2} p_i\right) - \frac{1}{\sum_{i=t_1}^{t_2} p_i}\left[p_{t_1} \log_e p_{t_1} + \left(\sum_{i=t_1+1}^{t_2-1} p_i\right) \times \log_e\left(\sum_{i=t_1+1}^{t_2-1} p_i\right) + p_{t_2} \log_e p_{t_2}\right] \\ &\vdots \\ S_i\left(M_i\right) &= \log_e\left(\sum_{i=t_{i-1}}^{t_i} p_i\right) - \frac{1}{\sum_{i=t_{i-1}}^{t_i} p_i}\left[p_{t_{i-1}} \log_e p_{t_{i-1}} + \left(\sum_{i=t_{i-1}+1}^{t_i-1} p_i\right) \times \log_e\left(\sum_{i=t_{i-1}+1}^{t_i-1} p_i\right) + p_{t_i} \log_e p_{t_i}\right] \\ &\vdots \\ S_{K+1}\left(M_{K+1}\right) &= \log_e\left(\sum_{i=t_K}^{L-1} p_i\right) - \frac{1}{\sum_{i=t_K}^{L-1} p_i}\left[p_{t_K} \log_e p_{t_K} + \left(\sum_{i=t_K+1}^{L-1} p_i\right) \log_e\left(\sum_{i=t_K+1}^{L-1} p_i\right)\right] \end{aligned} \tag{6}$$

The optimal threshold values $\left(t_1^*, t_2^*, \dots, t_K^*\right)$ are obtained by minimizing the interdependency using the relation

$$\left(t_1^*, t_2^*, \dots, t_K^*\right) = \arg \min_{0<t_1^*<t_2^*<\cdots<t_K^*<L-1} \left\{ S_1\left(M_1\right) + S_2\left(M_2\right) + \cdots + S_{K+1}\left(M_{K+1}\right) \right\}. \tag{7}$$

Need to mention that Eq. (7) is proposed here as the objective function, which is to be minimized by an optimizer. The idea is very intriguing, because it takes the shred boundary into consideration. This section shows the originality over other methods. To be specific, this kind of minimization function was never used (as per our knowledge) for the multilevel image thresholding application. The shred boundary between various classes are minimized with simpler calculations, which is interpreted from the Eqs. (6) and (7) . Further analysis and interpretations are found in the results and discussion section.

3. The proposed adaptive equilibrium optimizer (AEO)

In this section, we propose an adaptive equilibrium optimizer (AEO), which is another contribution of the work. This is an advancement to the equilibrium optimizer (EO) (Faramarzi et al., 2020). The originality of the proposal is based on random dispersal of nonperformer particles (search agents) in the search space, which is decided adaptively from the fitness value. This feature makes it more attractive for optimization. The EO is inspired by the dynamic and equilibrium state of the control volume mass balance model, based on the physics principle of mass conservation during entering, leaving, and generating in a control volume.

3.1. Mathematical modeling of AEO

Let the search agents are related to the concentration in a search space and initialized at iteration $iter = 1$ as follows:

$$C_i\,(iter = 1) = LB + rand_i\,(1, d) * (UB - LB),\ i = 1, 2, \ldots, N \tag{8}$$

The LB and UB are the lower and the upper bound of the search space, N is the number of search agents, d is the dimension of the problem and $rand_i$ is a one-dimensional vector consisting of the random numbers in the range of $[0, 1]$. The position of the ith search agent for a control volume V in EO is updated as

$$\vec{C}_i\,(new) = \vec{C}_{eq}\,(iter) + \left(\vec{C}_i\,(iter) - \vec{C}_{eq}\,(iter)\right) * \vec{F}_i\,(iter) + \frac{\vec{G}_i\,(iter)}{\vec{\lambda}_i\,(iter) * V} \times \left(1 - \vec{F}_i\,(iter)\right). \tag{9}$$

The $\vec{C}_{eq}$ is the randomly chosen equilibrium candidate from an equilibrium pool $\vec{C}_{eq,pool}$ of four best search agents $\vec{C}_{eq(1)}$, $\vec{C}_{eq(2)}$, $\vec{C}_{eq(3)}$, $\vec{C}_{eq(4)}$; and the average of these four search agents is $\vec{C}_{eq(ave)}$. The $\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}$ are decided using their fitness values $fit\left(\vec{C}_{eq(1)}\right)$, $fit\left(\vec{C}_{eq(2)}\right)$, $fit\left(\vec{C}_{eq(3)}\right)$ and $fit\left(\vec{C}_{eq(4)}\right)$. For a minimization problem, the equilibrium candidates and their fitness values are decided with the help of a sorted list. The fitness values of all N search agents are given as:

$$fit = \left(fit_1, fit_2, \ldots, fit_N\right) \tag{10}$$

These values are arranged in ascending order:

$$[sorted\_fit, sort\_index] = sort\,(fit) \tag{11}$$

Then, the equilibrium candidates and their fitness are described as:

$$\begin{aligned} fit\left(\vec{C}_{eq(1)}\right) &= sorted\_fit\,(1) \text{ and } \vec{C}_{eq(1)} = \vec{C}\,(sort\_index\,(1)) \\ fit\left(\vec{C}_{eq(2)}\right) &= sorted\_fit\,(2) \text{ and } \vec{C}_{eq(2)} = \vec{C}\,(sort\_index\,(2)) \\ fit\left(\vec{C}_{eq(3)}\right) &= sorted\_fit\,(3) \text{ and } \vec{C}_{eq(3)} = \vec{C}\,(sort\_index\,(3)) \\ fit\left(\vec{C}_{eq(4)}\right) &= sorted\_fit\,(4) \text{ and } \vec{C}_{eq(4)} = \vec{C}\,(sort\_index\,(4)) \\ \vec{C}_{eq(ave)} &= \frac{1}{4}\left(\vec{C}_{eq(1)} + \vec{C}_{eq(2)} + \vec{C}_{eq(3)} + \vec{C}_{eq(4)}\right) \end{aligned} \tag{12}$$

Finally, the equilibrium pool is expressed as

$$\vec{C}_{eq,pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\}. \tag{13}$$

The exponential term $\vec{F}_i$ to assist the EO for the exploitation and the exploration is calculated for the ith search agent as

$$\vec{F}_i\,(iter) = a_1\,sign\left(r_1 - 0.5\right)\left[e^{-\vec{\lambda}_i\left(1 - \frac{iter}{max\_iter}\right)^{\left(a_2\,\frac{iter}{max\_iter}\right)}}\right]. \tag{14}$$

Note that a_1 is used to control the exploration, a_2 is used to control the exploitation, $sign$ is to control the direction of search depending on a random number r_1 in between $[0, 1]$, $\vec{\lambda}_i\,(iter)$ is a random vector of d dimension in the interval $[0, 1]$ for the ith search agent in $iter$ iteration, $iter$ is the current iteration and max_iter is the maximum number of iterations the EO will go through in the position update.

The generation rate $\vec{G}_i$ help in the exploration using the participation probability of $\vec{C}_{eq}$. The $\vec{G}_i$ is defined as

$$\vec{G}_i\,(iter) = \vec{G}_{i,0}\,(iter) * \vec{F}_i\,(iter). \tag{15}$$

The $\vec{G}_{i,0}\,(iter)$ and $\overrightarrow{GCF}_i\,(iter)$ are evaluates as

$$\vec{G}_{i,0}\,(iter) = \overrightarrow{GCF}_i\,(iter)\left(\vec{C}_{eq}\,(iter) - \vec{\lambda}_i\,(iter)\right) \tag{16}$$

$$\overrightarrow{GCF}_i\,(iter) = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{17}$$

where GCF is the generation rate control factor and GP is the generation probability, r_1 and r_2 are the random numbers in the interval $[0, 1]$.

It is observed that the position of the search agents in the next iteration always depends on the equilibrium pool, which consists of four best positions and an average among the best post position. This narrows the boundary of the search space during an exploration in each progressive iteration, which leads to the search agent falling into a local minimum. To overcome this problem, we suggest an adaptive equilibrium optimizer (AEO), in which search agents go for a random dispersal for a nonperformer. An adaptive decision is taken with the help of the current fitness of the search agents and the average fitness of all N search agents. This is mathematically modeled for the minimization problem is defined as

$$\vec{C}_i\,(iter + 1) = \begin{cases} \vec{C}_i\,(new) & fit_i\,(iter) < fit_{avg}\,(iter) \\ \vec{C}_i\,(new) \otimes (0.5 + rand\,(1, d)) & fit_i\,(iter) \geq fit_{avg}\,(iter) \end{cases} \tag{18}$$

where

$\otimes$ represents the element-wise multiplication.

$fit_i\,(iter)$ represents the fitness value of the ith search agent at an iteration $iter$.

$fit_{avg}\,(iter)$ represents the average fitness of all search agents at an iteration $iter$, and is calculated as

$$fit_{avg}\,(iter) = \frac{1}{N}\sum_{i=1}^{N} fit_i\,(iter). \tag{19}$$

The AEO also inherits the concepts of memory saving form EO, in which the fitness value of the current iteration and the previous iteration are compared and updated if it achieves a better fitness value. This is expressed as

$$\vec{C}_i\,(iter) = \begin{cases} \vec{C}_i\,(iter) & iter > 1 \text{ and } fit_i\,(iter) < fit_i\,(iter - 1) \\ \vec{C}_i\,(iter - 1) & iter > 1 \text{ and } fit_i\,(iter) \geq fit_i\,(iter - 1) \\ \vec{C}_i\,(iter) & iter = 1 \end{cases} \tag{20}$$

and

$$fit_i\,(iter) = \begin{cases} fit_i\,(iter) & iter > 1 \text{ and } fit_i\,(iter) < fit_i\,(iter - 1) \\ fit_i\,(iter - 1) & iter > 1 \text{ and } fit_i\,(iter) \geq fit_i\,(iter - 1) \\ fit_i\,(iter) & iter = 1 \end{cases} \tag{21}$$

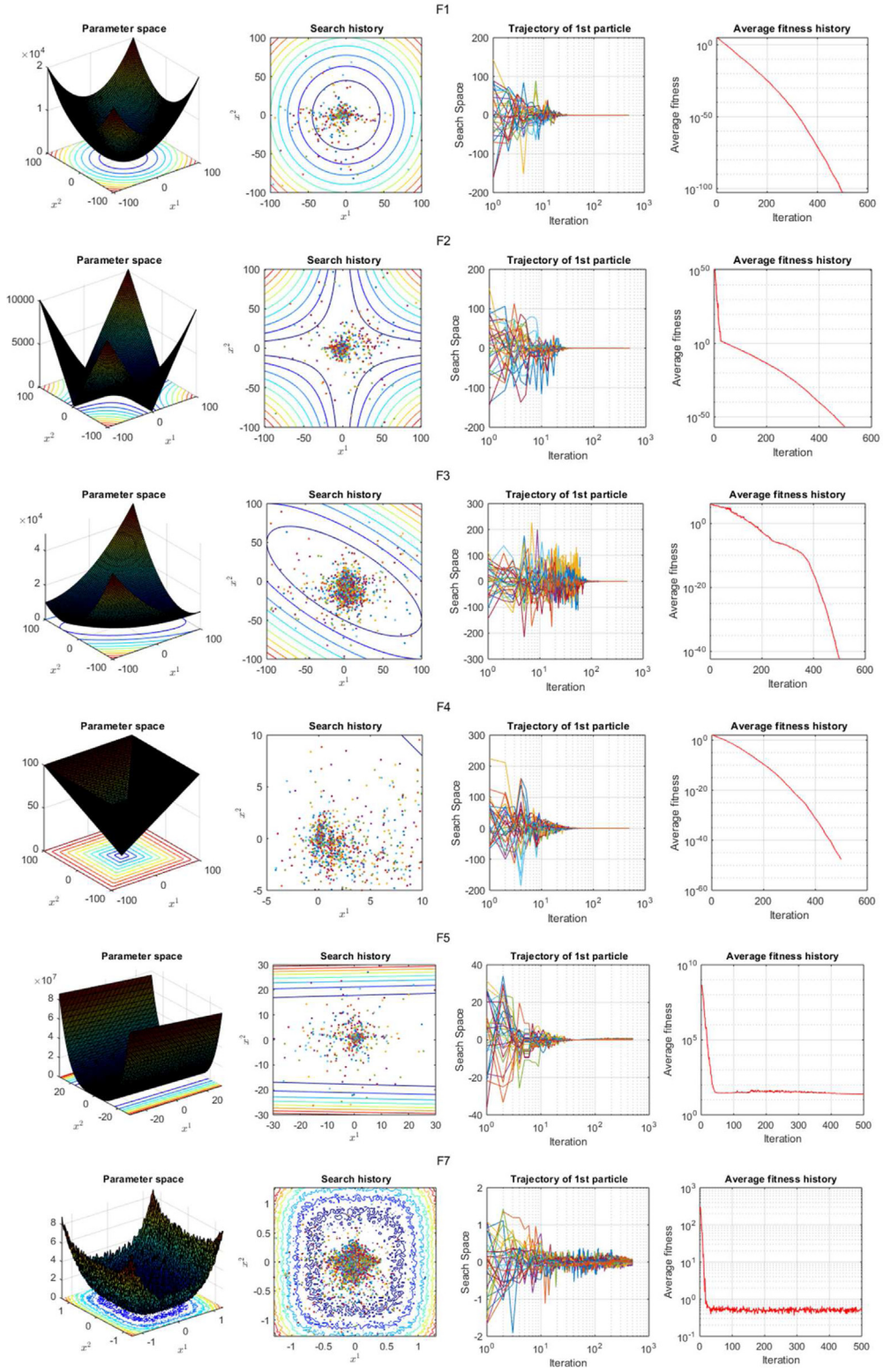

Fig. 1. Qualitative metric for unimodal benchmark functions F1, F2, F3, F4, F5, F6, and F7.

3.2. Pseudocode of the AEO

In the beginning, assign the number of the search agents N, maximum number of iterations max_iter, search dimension d, and the free parameters a_1, a_2, GP, V.

3.3. Performance evaluation of the AEO

This section describes the performance of the AEO, compared with the EO (Faramarzi et al., 2020). We have taken a set of 23 well-known benchmark functions from the literature (Liang and Suganthan, 2005;

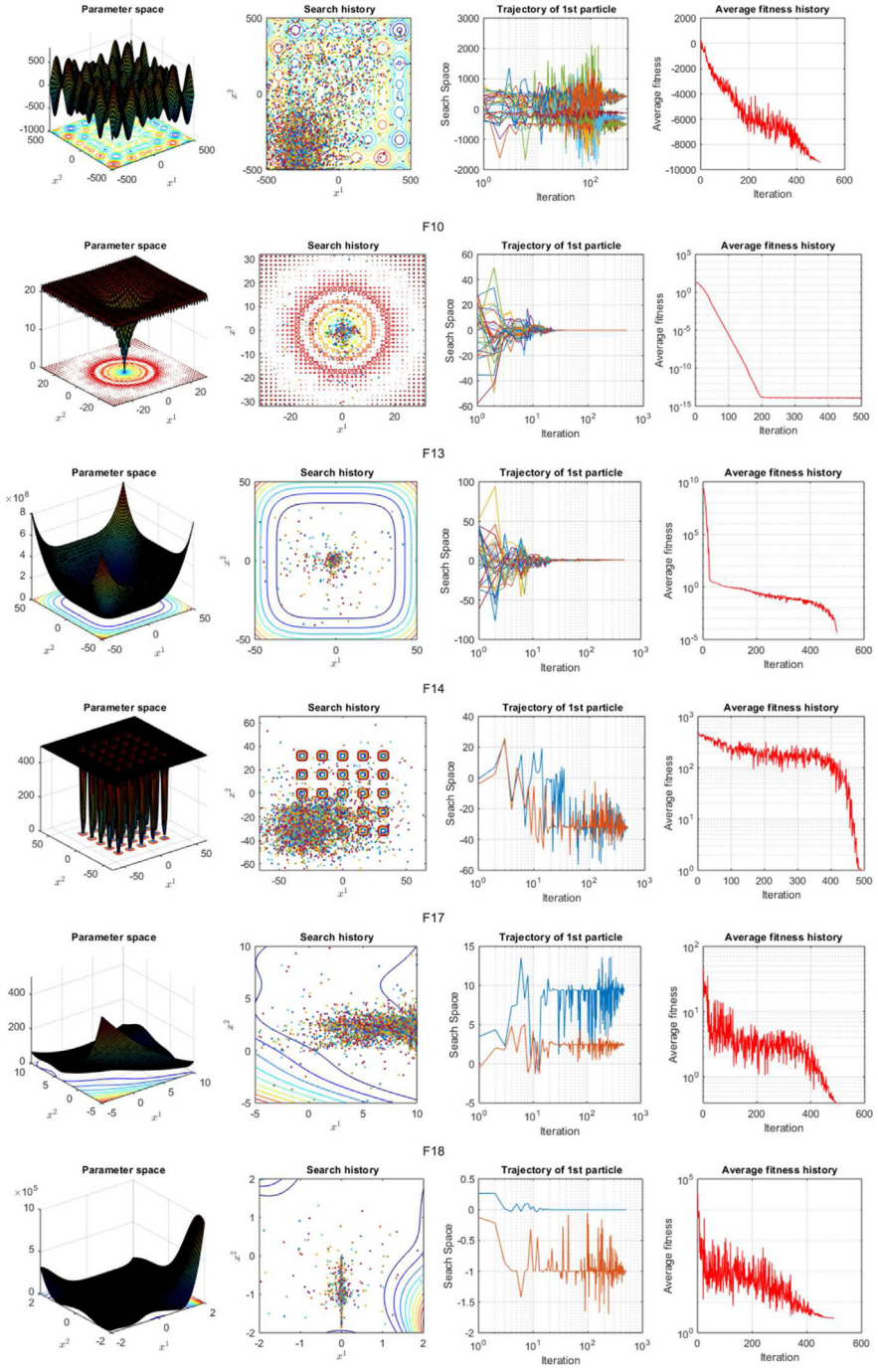

Fig. 2. Qualitative metric for multimodal benchmark functions F8, F10, F13, F14, F17, and F18.

Naik and Panda, 2016; Yao et al., 1999) to investigate the performance of the AEO. The benchmark functions are divided into three groups such as unimodal benchmark functions with scalable dimension (F1–F7), multimodal benchmark functions with scalable dimension (F8–F13), and multimodal benchmark functions with fixed dimension (F14–F23). The unimodal benchmark functions have unique minima, so it helps to recognize the exploitation performance of the algorithm, whereas the multimodal benchmark functions have many local minima, so it helps to recognize the exploration performance of the algorithm by avoiding local minima to reach the global minima. The performance of the AEO is compared with some recent promising optimization algorithms — EO (Faramarzi et al., 2020), GWO (Mirjalili et al., 2014), WOA (Mirjalili and Lewis, 2016), SSA (Jain et al., 2019), and WDO (Bayraktar et al., 2013). The qualitative measure comprised of the search history, trajectory, average fitness, convergence curve, and boxplot. Whereas the quantitative measure comprises of the average '*Ave*' and standard deviation '*Std*' values of the specific benchmark functions along with the convergence curves (the best fitness vs. iteration).

To maintain the consistency among the optimization algorithms, we have taken $N = 30$ and $max\_iter = 500$. For the scalable benchmark functions (F1–F13), the scalable dimensions are taken as $d = 10, 20, 30, 60, 100, 150, 200, 300, 500$ for providing a quantitative performance measure. Each benchmark function goes through 30 independent runs to record and compare the performances. The parameters used for the experiments are shown in Table 2

In the beginning, assign the number of the search agents N, maximum number of iterations max_iter, search dimension d, and the free parameters a_1, a_2, GP, V.

Initialization: Generate the random position vector C_i of the ith search agents using Eq. (8) for the N search agents for the iteration $iter = 1$.
For $iter = 1: max\_iter$
  Evaluate the fitness value fit for the current iteration.
  Find the equilibrium candidates $\vec{C}_{eq(1)}$, $\vec{C}_{eq(2)}$, $\vec{C}_{eq(3)}$, $\vec{C}_{eq(4)}$, and $\vec{C}_{eq(ave)}$ using the Eq. (12).
  Construct the equilibrium pool $\vec{C}_{eq,pool}$ using the Eq. (13).
  Accomplish memory saving using the Eq. (20) and Eq. (21).
  For $i = 1: N$
    Randomly choose the $\vec{C}_{eq}$ from the equilibrium pool $\vec{C}_{eq,pool}$.
    Construct the exponential term $\vec{F}_i$ using the Eq. (14).
    Construct the generation rate $\vec{G}_i$ using the Eq. (15).
    Calculate the average fitness fit_{avg} using the Eq. (19).
    Calculate the position search agent $\vec{C}_i(new)$ using the Eq. (9).
    Update the position of search agents $\vec{C}_i$ for the next iteration using the Eq. (18).
  End (For)
End (For)
Return the best solution as $\vec{C}_{eq(1)}$, and its best fitness as $fit(\vec{C}_{eq(1)})$.

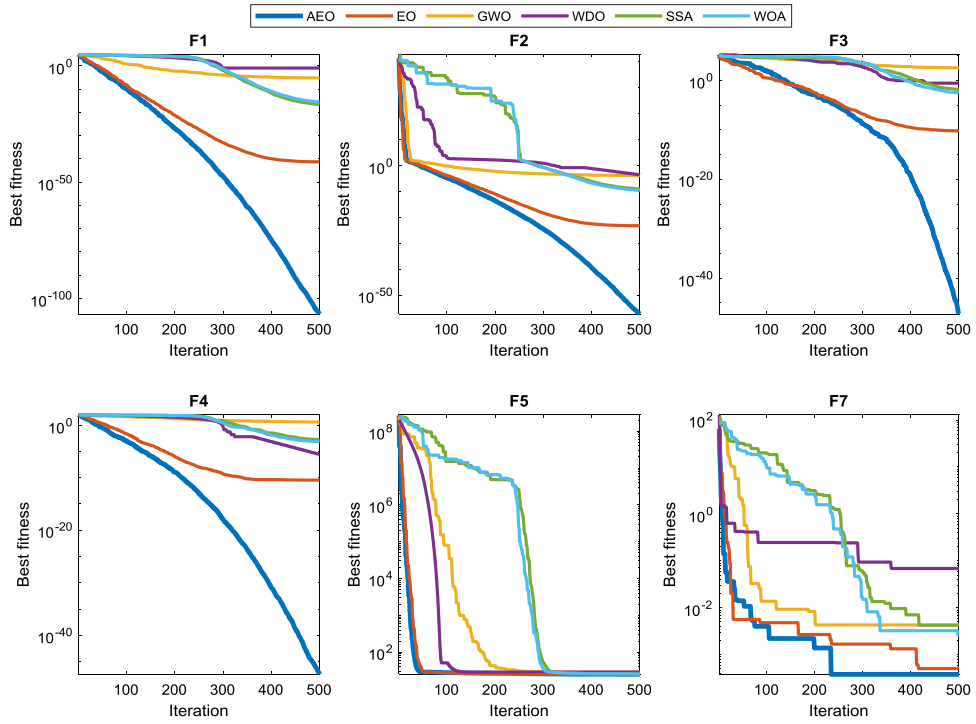

Fig. 3. Convergence curves of sample unimodal benchmark functions ($d = 30$).

3.3.1. Qualitative analysis (search history, trajectory, and average fitness) of AEO

The qualitative analysis of the several unimodal benchmark functions is presented in Fig. 1, and multimodal benchmark functions are presented in Fig. 2 of the AEO, which comprise of metrics: search history, trajectory, and average fitness history. The search history metric shows the position of all particles in the first two dimensions within the first iteration to the maximum iterations. From the second column of Figs. 1 and 2, it is visualized that a higher concentration of positions is aggregated nearer to the optimal solutions space more effectively during the unimodal benchmark functions than the multimodal benchmark functions. The trajectory metric shows the diversity of 1st particle in search space for a d dimensional problem. The trajectory curve presented in the third column of Figs. 1 and 2 reveals that, in the initial generation the positions of particle start with a large variation that covers the whole search space, however, as the optimization progresses, the positions of the particle converge to the solution space by following an oscillatory behavior. The average fitness history metric with a decreasing trend (minimization problem) or an increasing trend (maximization problem) shows how the optimizer uses the collaborative approach of particles to update their positions to reach the solution space. The average fitness history curve in the fourth column of Figs. 1 and 2 with a decreasing trend shows the stability. Based on the search history, trajectory and average fitness history metrics reveal that the AEO uses the exploration and the exploitation simultaneously more effectively to reach the optimal solutions.

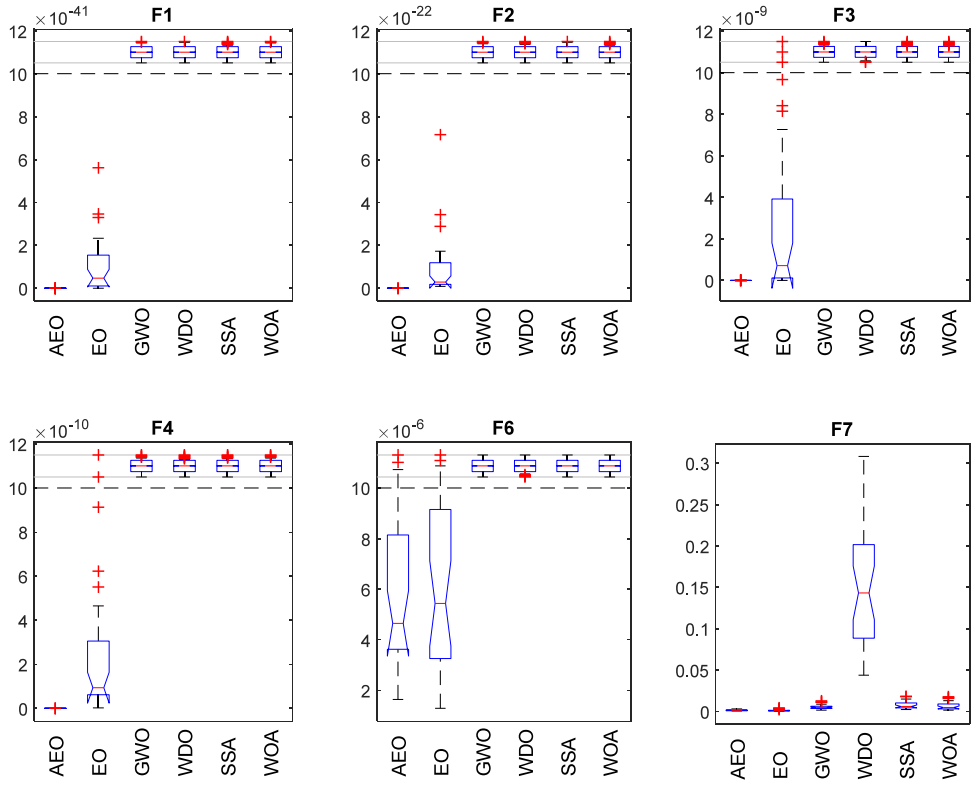

Fig. 4. Boxplot of sample unimodal benchmark functions ($d = 30$).

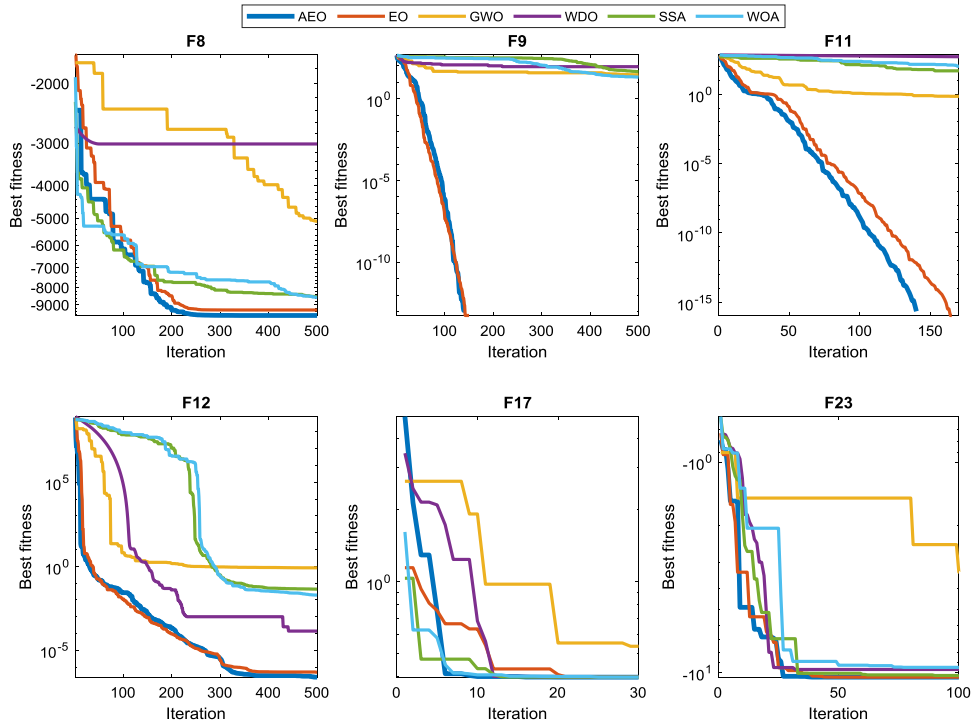

Fig. 5. Convergence curves of sample multimodal benchmark functions ($d = 30$ for F8, F9 and F10).

3.3.2. AEO performance on the unimodal benchmark functions

The unimodal benchmark functions have unique minima and explore the exploration capability of the algorithm. Statistical results are presented in Table 3, and the sample convergence curves of the benchmark functions are displayed in Fig. 3 (for $d = 30$). For the benchmark functions F1–F4, the AEO outperforms the EO in both the average and standard deviation values. For the other benchmark functions F5–F7, the AEO exhibits noteworthy differences in the average and standard deviation values, it performs well. The convergence curves presented in Fig. 3 also explains that the AEO has a better exploration than the EO, to get the optimal minimum. To understand the performance of the AEO vs. other algorithms over 30 independent

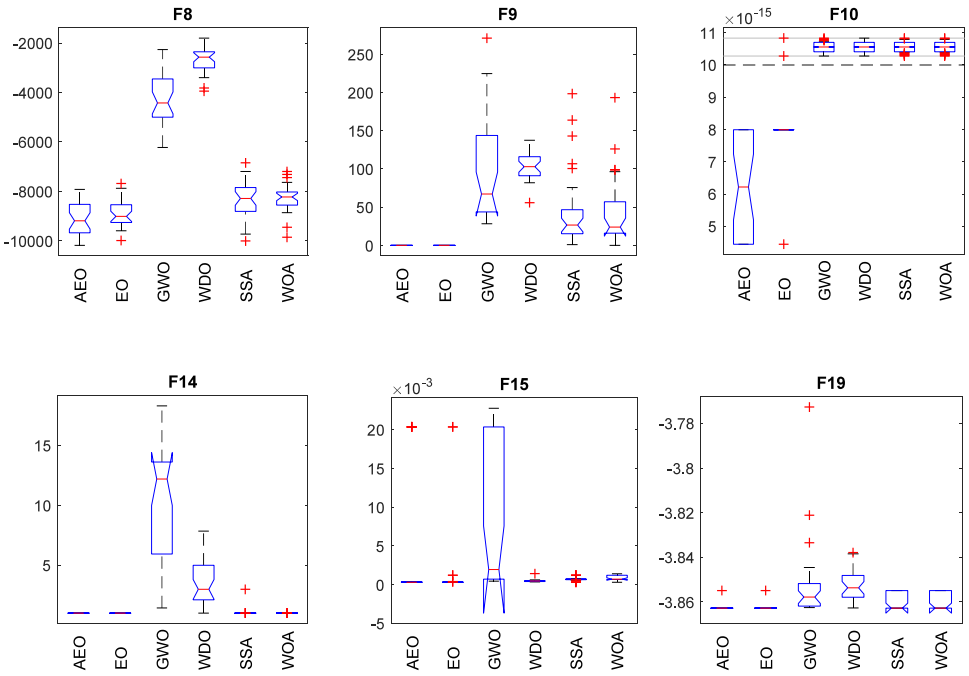

Fig. 6. Boxplot of sample multimodal benchmark functions ($d = 30$ for F8, F9, and F10).

Table 2
Parameter settings.

Algorithm	Parameters	Values
AEO/EO	Control the exploration (a_1)	2
	Control the exploitation (a_2)	1
	Generation probability (GP)	0.5
GWO	a	[2, 0]
WDO	RT coefficient	5
	Gravitational constant	0.2
	Coriolis effect	0.4
	Maximum allowed speed	0.3
SSA	Gliding constant (G_c)	1.9
	Scaling factor (sf)	18
	Predator presence probability (P_{dp})	0.1
WOA	a	[2, 0]
	b	1
	l	[−1, 1]

runs, a boxplot of optimal fitness is presented in Fig. 4, which reveals that AEO is more consistent among all optimization algorithms. The Friedman mean rank for benchmark functions F1–F7 (with $d = 10, 20, 30, 60, 100, 150, 200, 300, 500$) are evaluated based on the results obtained and presented in Table 3. It reveals that the AEO ranked first for the unimodal scalable benchmark functions. From this, it is implicit that the AEO has a better exploration ability.

3.3.3. The AEO performance on the multimodal benchmark functions

The multimodal benchmark functions have many local minima and explore the exploitation capability of the algorithm, we present the statistical result of scalable dimension in Table 4 and the fixed dimension in Table 5. Moreover, the AEO outperforms for multimodal benchmark functions F8 to F11, except F12 and F13 where AEO performs below WDO. The convergence curve of sample benchmark functions is presented in Fig. 5, which shows the AEO and EO perform similarly in multimodal benchmark functions. Still, AEO shows a little faster in convergence than EO, and quite more for GWO, WDO, SSA, and WOA. The distribution of results obtained from 30 independent runs is shown using a boxplot in Fig. 6, which reveals AEO is more consistent among all presented optimization techniques to get optimal solutions. The Friedman mean rank for multimodal benchmark functions F8–F13 and F14–F23 ranked first among all presented optimization algorithms. From this, it is implicit that the AEO has a better exploitation ability.

3.3.4. Scalability analysis

This section presents a scalability assessment to investigate the influence of dimensions on the optimal results of the AEO. The experiments are performed with 10, 20, 30, 60, 100, 150, 200, 300, and 500 dimensions on the scalable unimodal and multimodal benchmark functions F1–F13. The average and the statistical results of scalable benchmark functions presented in Tables 3 and 4, and a graphic representation of the average results are shown in Fig. 7. If we analyze Tables 3 and 4 and Fig. 7, the AEO outperforms in most of the test cases benchmark functions, such as F1–F4, F6, F9–F11. However, the AEO performs a similar way in F5, F7–8, F12–13 with the EO, and the AEO lags WDO for the test cases benchmark functions F12–13. To summarize the comparisons, we perform Friedman mean rank test on the average value and presented in Fig. 8. Results reveal that the AEO ranked first in all dimensions.

3.4. Discussion on results of AEO

As per the qualitative and the quantitative analysis discussed in the previous section, the AEO has shown an improvement over its predecessors EO, and superiority over some recent successful optimizers like GWO, WDO, WOA, and SSA. The AEO can be used for a wide range of optimization problems, as it has shown less impact when dimensions change from low dimensions to high dimensions, that is revealed from Tables 3 and 4, and Figs. 7 and 8. Although the most of the optimization problems shown similar results in multimodal benchmark functions with fixed dimensions, still the AEO has achieved improvement when compared with another optimizer. The Friedman's mean rank outcome from average fitness of 30 independent runs for 500 iterations with 30 particles (population) are presented in Tables 3–5 and Fig. 8, which marked the AEO as first ranked in all possible

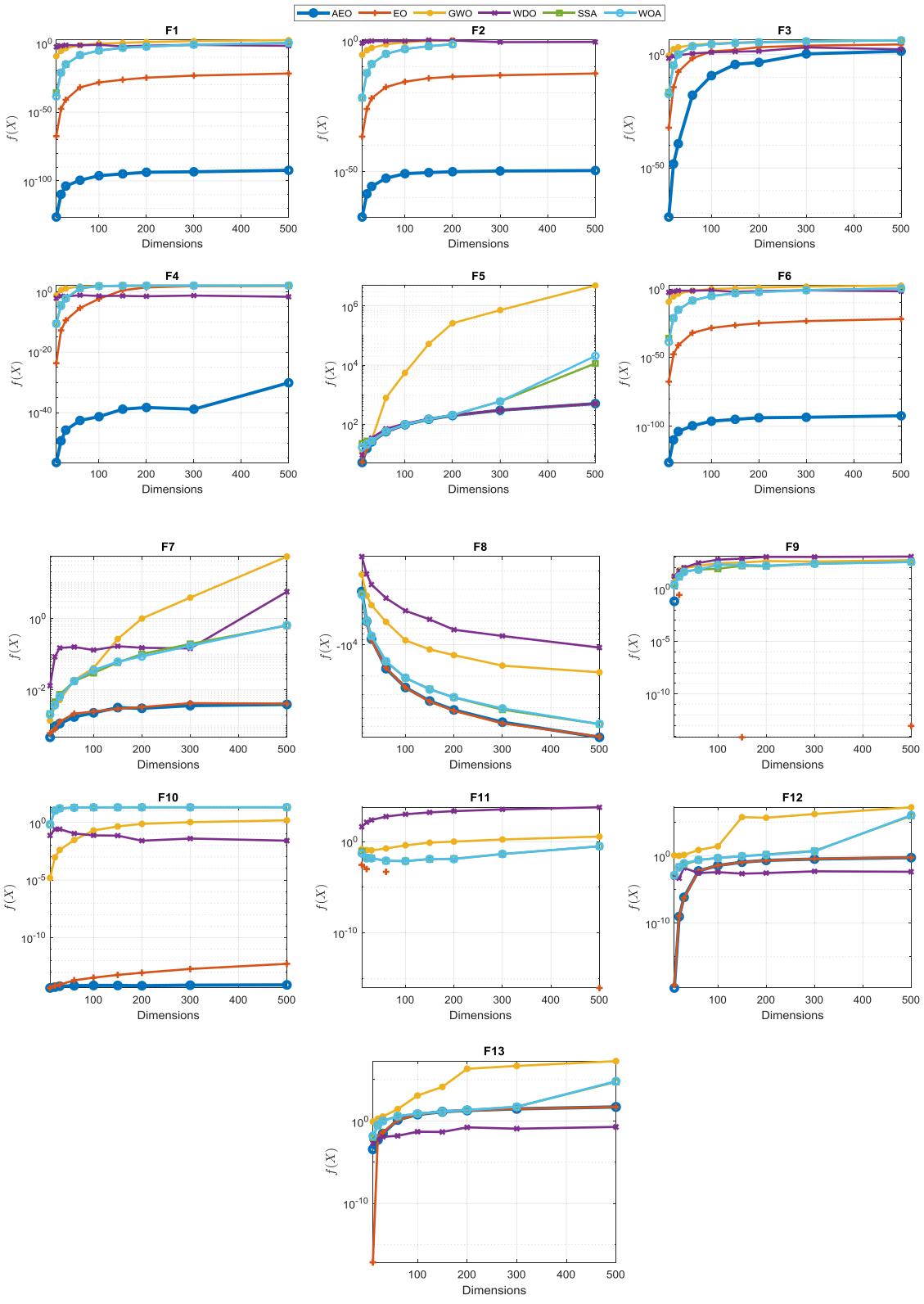

Fig. 7. Scalability results from comparisons of scalable unimodal and multimodal benchmark functions F1–F13.

test cases like the unimodal benchmark functions (F1–F7), the multimodal benchmark functions with scalable dimensions (F8–F13), the multimodal benchmark functions with fixed dimensions (F14–F23) and scalability analysis. Finally, the AEO has overcome the problem of the EO with a random dispersal to nonperformer search agents positions. It is reiterated that the AEO decides the dispersal of a nonperformer by adaptively from the current and the previous fitness values. This helps the AEO to maintain a good balance between the exploration and the exploitation to reach optimal solutions.

Table 3
Statistical results and comparison of the unimodal benchmark functions with the scalable dimensions.

Function	d	AEO		EO		GWO		WDO		SSA		WOA	
		Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1 (Sphere model)	10	**4.5316E−128**	**1.2329E−127**	2.5568E−67	9.2181E−67	5.1084E−10	1.4067E−09	3.3229E−03	8.5025E−03	1.3385E−36	7.3044E−36	2.9497E−39	9.9167E−39
	20	**1.4971E−110**	**7.4082E−110**	2.8626E−48	5.1861E−48	4.1337E−06	1.2062E−05	1.7855E−02	3.6388E−02	5.1652E−22	1.3330E−21	6.7521E−22	1.4549E−21
	30	**1.6176E−104**	**7.4697E−104**	1.0581E−41	1.3061E−41	3.8610E−04	9.9033E−04	6.0504E−02	8.8863E−02	5.4167E−16	1.2524E−15	7.5096E−16	1.2641E−15
	60	**2.7966E−100**	**7.1987E−100**	1.1877E−32	2.7083E−32	5.0168E−02	1.2317E−01	5.1678E−02	9.6092E−02	3.5774E−09	6.2982E−09	3.9021E−09	3.6575E−09
	100	**1.4718E−97**	**4.9262E−97**	5.7333E−29	9.3923E−29	9.0294E−01	1.1478E+00	1.1722E−01	3.3193E−01	6.7479E−06	4.6122E−06	9.2337E−06	9.9596E−06
	150	**1.2204E−95**	**5.0463E−95**	4.0602E−27	4.4389E−27	2.7766E+00	3.0093E+00	8.0352E−03	2.5388E−02	7.7610E−04	1.0491E−03	6.9700E−04	8.0738E−04
	200	**1.8336E−94**	**5.1294E−94**	1.2680E−25	1.1855E−25	1.2327E+01	1.4356E+01	2.7509E−02	4.8174E−02	6.3053E−03	4.2312E−03	5.6752E−03	3.3776E−03
	300	**4.0213E−94**	**9.1279E−94**	4.4098E−24	5.6717E−24	5.9041E+01	6.3804E+01	9.3741E−02	4.3170E−01	1.4021E−01	9.0584E−02	1.2671E−01	1.0283E−01
	500	**3.5620E−78**	**1.4821E−77**	1.4974E−19	1.6282E−19	3.1166E+02	2.8138E+02	2.1659E−02	3.2244E−02	2.6574E+00	1.3959E+00	3.0701E+00	2.0514E+00
F2 (Schwefel's problem 2.22)	10	**2.6834E−67**	**6.9476E−67**	1.0145E−36	2.5460E−36	3.5171E−06	6.7766E−06	1.0597E−01	1.5820E−01	1.2626E−22	3.5153E−22	1.6237E−22	2.9070E−22
	20	**2.8542E−59**	**9.3956E−59**	7.9549E−27	7.8606E−27	2.0630E−04	2.4654E−04	4.5465E−01	4.6465E−01	5.1613E−13	7.0630E−13	3.4231E−13	3.0351E−13
	30	**1.3763E−56**	**3.6496E−56**	6.8708E−23	6.3923E−23	1.6601E−03	2.0888E−03	6.9059E−01	7.7976E−01	1.0896E−09	1.1934E−09	1.1282E−09	1.1825E−09
	60	**2.6816E−53**	**7.8539E−53**	1.7674E−18	1.3208E−18	3.3874E−02	3.4720E−02	5.5575E−01	7.2388E−01	7.5341E−06	4.5786E−06	8.7926E−06	5.9859E−06
	100	**1.9910E−52**	**3.3578E−52**	2.9862E−16	3.5418E−16	2.3522E−01	1.3212E−01	6.6586E−01	8.8375E−01	6.1103E−04	2.5882E−04	5.1440E−04	2.2165E−04
	150	**4.0437E−51**	**8.7123E−51**	4.2037E−15	2.8741E−15	7.1735E−01	3.8017E−01	1.5355E+00	1.7385E+00	6.9160E−03	2.0356E−03	7.8144E−03	2.8995E−03
	200	**8.4881E−51**	**3.5927E−50**	1.5758E−14	1.1305E−14	1.3906E+00	7.0255E−01	1.0684E+00	9.0240E−01	3.7441E−02	1.2811E−02	3.9092E−02	1.3885E−02
	300	**1.5588E−50**	**2.6408E−50**	5.7519E−14	3.8077E−14	Inf	NaN	2.4747E−01	3.3736E−01	Inf	NaN	Inf	NaN
	500	**3.6954E−43**	**9.0843E−43**	1.2469E−11	8.5066E−12	Inf	NaN	2.9763E−01	2.7871E−01	Inf	NaN	Inf	NaN
F3 (Schwefel's problem 1.2)	10	**2.5216E−68**	**1.3804E−67**	2.2123E−32	7.4651E−32	3.0891E−01	7.4527E−01	2.6508E−02	1.6525E−02	1.9823E−17	7.0295E−17	3.5277E−18	9.0570E−18
	20	**6.1403E−49**	**3.3622E−48**	4.0193E−15	1.7044E−14	2.5682E+02	4.2221E+02	2.0069E−01	9.7281E−02	2.8598E−05	8.1933E−05	1.5580E−05	3.5330E−05
	30	**4.2442E−38**	**2.3009E−37**	2.7162E−09	1.0977E−08	1.6541E+03	2.2046E+03	5.0269E−01	2.2895E−01	1.8543E+00	5.7429E+00	1.0422E+00	2.4691E+00
	60	**1.2351E−18**	**5.1367E−18**	2.3321E−02	4.9967E−02	1.6802E+04	1.0276E+04	2.9492E+00	9.9418E−01	4.9241E+03	4.7592E+03	4.4956E+03	5.4226E+03
	100	**4.0057E−09**	**1.9844E−08**	2.9897	7.4947	7.2166E+04	2.9155E+04	9.6933E+00	5.3503E+00	3.8678E+04	1.7309E+04	3.9996E+04	2.1330E+04
	150	**4.6219E−05**	**1.7824E−04**	1.1369E+02	2.0820E+02	1.8282E+05	8.1588E+04	1.9061E+01	7.3273E+00	1.3111E+05	5.0500E+04	1.3124E+05	4.5275E+04
	200	**3.3762E−04**	**1.6857E−03**	1.8027E+03	4.1867E+03	3.5535E+05	1.2849E+05	3.1446E+01	9.6460E+00	2.6221E+05	8.6973E+04	2.6220E+05	7.6059E+04
	300	**2.0010E+00**	**1.0683E+01**	8.7442E+03	1.0751E+04	8.1465E+05	2.5414E+05	1.4032E+03	7.2415E+03	6.5477E+05	1.9045E+05	6.9466E+05	2.0119E+05
	500	**264.2696**	**777.6426**	5.5807E+04	5.2259E+04	2.4589E+06	5.9266E+05	2.1073E+02	1.3409E+02	1.8587E+06	4.9196E+05	1.9359E+06	4.1922E+05
F4 (Schwefel's problem 2.21)	10	**4.0835E−58**	**8.5382E−58**	1.1785E−24	2.1915E−24	4.7277E−02	8.5411E−02	6.4406E−03	2.0959E−02	2.8198E−11	9.0734E−11	2.2915E−11	6.2927E−11
	20	**4.5978E−50**	**7.8460E−50**	1.5123E−13	3.8708E−13	3.6040E+00	5.5432E+00	3.1747E−02	5.0612E−02	4.2940E−05	7.8481E−05	1.8814E−05	2.2233E−05
	30	**1.9382E−46**	**9.1426E−46**	1.9287E−10	2.0685E−10	1.1079E+01	8.4198E+00	2.4113E−02	5.9060E−02	5.7120E−03	1.0197E−02	5.0713E−03	8.7062E−03
	60	**2.5526E−43**	**7.7436E−43**	3.7825E−06	6.1035E−06	4.8503E+01	1.4377E+01	6.5580E−02	8.5759E−02	1.0281E+01	9.7119E+00	1.5436E+01	1.5235E+01
	100	**4.5302E−42**	**1.4249E−41**	0.0026	0.0057	6.8613E+01	8.7373E+00	3.5387E−02	6.6557E−02	6.9166E+01	1.1014E+01	6.5336E+01	1.4628E+01
	150	**1.3343E−39**	**6.8886E−39**	2.7564E+00	1.2760E+01	7.6819E+01	6.3137E+00	3.8671E−02	7.5310E−02	8.8649E+01	4.0413E+00	8.7372E+01	3.8829E+00
	200	**5.0219E−39**	**1.9689E−38**	2.3775E+01	2.9738E+01	8.0801E+01	5.2242E+00	2.8747E−02	7.4142E−02	9.3556E+01	1.4524E+00	9.2933E+01	2.7598E+00
	300	**1.3230E−39**	**4.4055E−39**	6.2908E+01	2.1770E+01	8.5304E+01	4.2162E+00	4.8115E−02	1.0063E−01	9.6089E+01	1.5464E+00	9.6042E+01	9.8502E−01
	500	**5.0507E−28**	**2.3494E−27**	75.6817	14.2782	8.8927E+01	3.7229E+00	1.9637E−02	6.2745E−02	9.8230E+01	4.8144E−01	9.7991E+01	6.8908E−01
F5 (Generalized Rosenbrock's function)	10	**5.0683**	0.1557	5.1564	**0.1463**	1.1102E+01	1.3406E+01	9.0502E+00	2.7065E−01	2.2254E+01	3.8294E+01	1.6470E+01	3.4752E+01
	20	**1.5198E+01**	**1.7091E−01**	1.5300E+01	2.0529E−01	1.8836E+01	2.3528E−01	2.2730E+01	3.3047E+00	2.7058E+01	3.5070E+01	2.0925E+01	2.5369E+01
	30	**25.3250**	**0.2357**	25.3666	0.2843	2.8989E+01	4.6666E−01	3.4311E+01	6.9136E+00	2.6658E+01	1.0116E+00	2.6944E+01	1.2324E+00
	60	**5.5660E+01**	**4.2432E−01**	5.5932E+01	7.9099E−01	7.7830E+02	1.9843E+03	7.0275E+01	1.8283E+01	5.7647E+01	8.2064E−01	5.7480E+01	9.9571E−01
	100	**96.0017**	**0.8157**	96.3462	0.8454	5.4554E+03	8.3235E+03	1.0496E+02	1.3446E+01	9.7651E+01	6.9779E−01	9.7830E+01	6.4219E−01
	150	**1.4687E+02**	**7.6849E−01**	1.4735E+02	1.0043E+00	5.1941E+04	9.5179E+04	1.5178E+02	6.6057E+00	1.4869E+02	4.0912E−01	1.4857E+02	1.0652E+00
	200	**1.9703E+02**	**7.7957E−01**	1.9717E+02	8.6248E−01	2.6035E+05	3.3267E+05	2.0127E+02	6.8933E+00	2.0506E+02	6.0856E+00	2.0504E+02	9.0256E+00
	300	**2.9726E+02**	5.4191E−01	2.9759E+02	3.6366E−01	7.1900E+05	9.8545E+05	3.0082E+02	6.3638E+00	5.9036E+02	2.5587E+02	5.9006E+02	3.0587E+02
	500	**497.6544**	**0.2157**	497.9012	0.2839	4.8829E+06	5.5241E+06	5.0207E+02	1.0662E+01	1.1364E+04	1.2938E+04	2.0353E+04	3.3220E+04
F6 (Step function)	10	**3.3364E−19**	**7.2602E−19**	5.6522E−19	1.3174E−18	1.5398E+00	3.2457E−01	2.2646E−02	1.4650E−02	1.7990E−05	8.6139E−06	8.3979E−03	4.5895E−02
	20	**1.4502E−08**	**2.9613E−08**	1.6467E−08	3.6224E−08	3.0761E+00	4.8710E−01	1.2865E−01	7.3607E−02	2.5413E−01	2.1790E−01	1.9850E−01	1.3692E−01
	30	**6.4507E−06**	**4.9800E−06**	9.0580E−06	5.6050E−06	5.2251E+00	5.1542E−01	3.3939E−01	1.8412E−01	1.0184E+00	3.2902E−01	9.6081E−01	4.6154E−01
	60	**2.9823E−01**	**1.9686E−01**	3.6494E−01	3.2941E−01	1.2151E+01	6.9909E−01	1.2765E+00	6.8445E−01	5.1198E+00	7.2415E−01	5.1897E+00	7.2566E−01
	100	**3.4933**	0.7013	3.8269	**0.6037**	2.2079E+01	1.4209E+00	2.5087E+00	2.2349E+00	1.3157E+01	8.8141E−01	1.2758E+01	7.9493E−01
	150	**1.0496E+01**	**7.7977E−01**	1.1375E+01	1.1878E+00	3.9199E+01	7.1292E+00	5.1378E+00	4.6399E+00	2.4022E+01	8.4490E−01	2.4047E+01	8.9284E−01
	200	**1.9124E+01**	**1.2049E+00**	2.1201E+01	1.3237E+00	5.5263E+01	8.8193E+00	1.0154E+01	8.3037E+00	3.6189E+01	1.0980E+00	3.5820E+01	9.6328E−01
	300	**3.8525E+01**	**1.6542E+00**	4.1265E+01	1.6760E+00	1.2566E+02	7.3821E+01	2.3610E+01	1.8125E+01	6.0154E+01	1.3365E+00	6.0766E+01	1.1043E+00
	500	**89.8830**	**1.9246**	94.6020	1.9554	4.2037E+02	2.3598E+02	6.0306E+01	3.5352E+01	1.1408E+02	3.2280E+00	1.1376E+02	3.3826E+00
F7 (quartic function i.e. noise)	10	**5.0627E−04**	**2.6553E−04**	5.4887E−04	4.3071E−04	1.4166E−03	8.4512E−04	1.3156E−02	5.4455E−03	2.1785E−03	1.7338E−03	2.0580E−03	1.5065E−03
	20	**8.1107E−04**	**5.2313E−04**	9.7291E−04	5.3133E−04	3.4759E−03	2.1234E−03	8.4674E−02	2.8876E−02	4.6615E−03	2.2990E−03	3.8072E−03	2.5942E−03
	30	**0.0011**	**5.8746E−04**	0.0013	8.7949E−04	5.2565E−03	2.7974E−03	1.4960E−01	7.3841E−02	7.4960E−03	4.4971E−03	6.5680E−03	4.6452E−03
	60	**1.7706E−03**	**8.1064E−04**	2.1920E−03	8.8486E−04	1.8822E−02	1.4451E−02	1.5858E−01	1.3293E−01	1.7639E−02	8.9648E−03	1.7781E−02	9.4189E−03
	100	0.0023	**8.4663E−04**	0.0023	9.8342E−04	4.1072E−02	2.4209E−02	1.3056E−01	1.2743E−01	2.9831E−02	1.3097E−02	3.6096E−02	1.5912E−02
	150	**2.9994E−03**	**1.3424E−03**	3.2257E−03	1.3433E−03	2.6492E−01	4.2684E−01	1.6668E−01	1.4212E−01	5.8456E−02	2.3114E−02	6.1403E−02	2.7317E−02
	200	**3.0900E−03**	1.1734E−03	3.3418E−03	**1.4411E−03**	9.8855E−01	1.4425E+00	1.5085E−01	1.5197E−01	1.0116E−01	4.4452E−02	8.6556E−02	3.1257E−02
	300	**3.6678E−03**	1.4375E−03	4.2898E−03	**1.4205E−03**	3.7849E+00	6.9597E+00	1.4468E−01	1.3637E−01	1.9301E−01	5.9751E−02	1.7110E−01	6.8587E−02
	500	**0.0048**	**0.0023**	0.0060	0.0026	5.2912E+01	1.0638E+02	5.5271E+00	2.9843E+01	6.2942E−01	2.7879E−01	6.4550E−01	3.4484E−01
Friedman mean rank		**1.1190**		2.1825		5.3968		4.0000		4.2143		4.0873	
Rank		**1**		2		6		3		5		4	

Table 4
Statistical results and comparison of the multimodal benchmark functions with the scalable dimensions.

Function	d	AEO		EO		GWO		WDO		SSA		WOA	
		Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F8 (Generalized Schwefel's problem 2.26)	10	**−3.2154E+03**	**279.9362**	−3.2052E+03	343.5169	−2.1691E+03	4.5415E+02	−1.4653E+03	2.8827E+02	−3.2678E+03	3.1858E+02	−3.3848E+03	3.1606E+02
	20	**−6.2893E+03**	**4.7436E+02**	−6.0277E+03	5.1587E+02	−3.4424E+03	6.4476E+02	−2.1444E+03	3.9810E+02	−5.9602E+03	5.7305E+02	−5.9901E+03	5.8047E+02
	30	**−8.9160E+03**	**621.1704**	−8.9108E+03	650.4205	−4.2359E+03	1.0159E+03	−2.7037E+03	5.1709E+02	−8.3456E+03	7.5921E+02	−8.2966E+03	5.6016E+02
	60	**−1.6984E+04**	1.1162E+03	−1.6649E+04	**9.4210E+02**	−6.1126E+03	1.4139E+03	−3.6347E+03	6.8416E+02	−1.4526E+04	6.8070E+02	−1.4478E+04	9.1612E+02
	100	**−2.5828E+04**	**1.3485E+03**	−2.5809E+04	1.6413E+03	−9.1387E+03	1.7121E+03	−4.7946E+03	9.3971E+02	−2.0841E+04	1.0755E+03	−2.0715E+04	7.3723E+02
	150	**−3.4955E+04**	**1.8236E+03**	−3.4479E+04	2.2652E+03	−1.1157E+04	2.2516E+03	−5.7916E+03	1.1259E+03	−2.6537E+04	9.8385E+02	−2.6843E+04	9.7697E+02
	200	**−4.2766E+04**	2.7479E+03	−4.1902E+04	**2.4796E+03**	−1.2648E+04	2.7262E+03	−7.2550E+03	1.2675E+03	−3.1918E+04	1.3167E+03	−3.1734E+04	1.5030E+03
	300	**−5.6228E+04**	**3.6189E+03**	−5.4752E+04	3.8690E+03	−1.5916E+04	3.5577E+03	−8.3155E+03	1.2499E+03	−4.1670E+04	1.7924E+03	−4.0501E+04	2.1503E+03
	500	**−7.6181E+04**	5.9297E+03	−7.5172E+04	**4.9373E+03**	−1.8393E+04	5.0405E+03	−1.0718E+04	2.0649E+03	−5.7150E+04	3.1258E+03	−5.6972E+04	3.2086E+03
F9 (Genreralized Rastrigin's function)	10	**0**	**0**	0.0633	0.2524	1.4893E+01	1.1324E+01	1.5916E+01	5.6597E+00	3.0243E+00	4.2833E+00	2.3181E+00	3.0917E+00
	20	**0**	**0**	2.6534E−01	1.4533E+00	5.7726E+01	4.2151E+01	5.1864E+01	1.2879E+01	1.3540E+01	1.4813E+01	1.5215E+01	1.6554E+01
	30	**0**	**0**	**0**	**0**	9.6151E+01	6.6492E+01	1.0413E+02	1.7393E+01	4.5404E+01	4.9585E+01	4.0932E+01	4.2997E+01
	60	**0**	**0**	**0**	**0**	1.5202E+02	1.0808E+02	2.9772E+02	2.0263E+01	6.9327E+01	8.7817E+01	6.8535E+01	6.5683E+01
	100	**0**	**0**	**0**	**0**	2.7578E+02	1.4824E+02	5.9477E+02	2.8804E+01	8.2262E+01	4.4332E+01	1.9388E+02	2.6281E+02
	150	**0**	**0**	7.5791E−15	4.1513E−14	3.1611E+02	3.1938E+02	7.6297E+02	4.1567E+02	1.5687E+02	1.1881E+02	1.8685E+02	3.1506E+02
	200	**0**	**0**	**0**	**0**	4.4012E+02	2.9626E+02	1.1219E+03	5.5472E+02	1.4326E+02	7.7690E+01	1.6347E+02	1.8034E+02
	300	**0**	**0**	**0**	**0**	3.9959E+02	2.2055E+02	1.1042E+03	1.1098E+03	2.4699E+02	1.4448E+02	2.4542E+02	1.6141E+02
	500	**0**	**0**	2.4253E−13	4.7369E−13	5.2832E+02	2.1900E+02	1.1872E+03	1.8201E+03	3.6508E+02	1.8663E+02	3.6035E+02	1.9400E+02
F10 (Ackley's function)	10	**4.4409E−15**	**0**	**4.4409E−15**	**0**	1.5024E−05	3.2373E−05	7.3746E−02	1.1688E−01	6.6386E−01	3.6361E+00	6.6344E−01	3.6338E+00
	20	**5.1514E−15**	**1.4454E−15**	6.8094E−15	1.7034E−15	9.2647E−04	2.1459E−03	2.5533E−01	2.2292E−01	1.0640E+01	1.0123E+01	1.0639E+01	1.0122E+01
	30	**5.9804E−15**	**1.7906E−15**	9.1778E−15	2.6933E−15	4.0940E−03	6.5420E−03	2.5800E−01	2.8368E−01	1.5964E+01	8.1183E+00	1.6623E+01	7.5612E+00
	60	**6.8094E−15**	**1.7034E−15**	2.0191E−14	4.4435E−15	2.9568E−02	3.4880E−02	1.0872E−01	1.5455E−01	2.0023E+01	4.3594E−02	1.8703E+01	5.0841E+00
	100	**6.8094E−15**	**1.7034E−15**	3.4994E−14	5.7210E−15	2.0122E−01	3.0773E−01	7.4798E−02	9.2893E−02	1.9482E+01	3.6803E+00	2.0158E+01	7.5603E−02
	150	**7.1646E−15**	**1.5283E−15**	5.8679E−14	1.1462E−14	4.4390E−01	4.1793E−01	7.0729E−02	8.8489E−02	2.0272E+01	7.1121E−02	2.0279E+01	6.1946E−02
	200	**6.8094E−15**	**1.7034E−15**	9.0417E−14	2.3070E−14	7.4395E−01	4.7259E−01	2.5721E−02	5.5426E−02	2.0334E+01	6.5235E−02	2.0340E+01	6.9929E−02
	300	**7.5199E−15**	**2.0298E−15**	1.9309E−13	8.9062E−14	1.0451E+00	4.9725E−01	3.9700E−02	4.8195E−02	2.0421E+01	5.7849E−02	2.0415E+01	5.7330E−02
	500	**7.5199E−15**	**2.0298E−15**	1.4498E−11	8.4054E−12	1.5099E+00	4.7520E−01	2.5946E−02	3.0686E−02	2.0478E+01	5.3609E−02	2.0483E+01	5.5186E−02
F11 (Generalized Griewank function)	10	**8.2165E−04**	**0.0025**	0.0066	0.0133	1.4413E−01	9.8746E−02	4.1536E+01	1.7944E+01	8.3089E−02	7.5019E−02	5.0243E−02	4.6532E−02
	20	**0**	**0**	9.8589E−04	3.9691E−03	1.1596E−01	9.8018E−02	1.2849E+02	2.6187E+01	1.8292E−02	2.1325E−02	1.4796E−02	1.3319E−02
	30	**0**	**0**	**0**	**0**	1.1051E−01	9.5677E−02	2.2792E+02	3.7495E+01	1.5486E−02	2.2904E−02	1.4891E−02	1.5833E−02
	60	**0**	**0**	4.9419E−04	2.7068E−03	1.7565E−01	1.7175E−01	5.4450E+02	5.9104E+01	7.4902E−03	9.8989E−03	8.0005E−03	1.1314E−02
	100	**0**	**0**	**0**	**0**	3.8520E−01	2.9057E−01	9.7253E+02	8.3378E+01	7.2718E−03	1.1193E−02	7.3252E−03	1.3380E−02
	150	**0**	**0**	**0**	**0**	7.5865E−01	3.0555E−01	1.5513E+03	9.4995E+01	1.1707E−02	1.6291E−02	1.2740E−02	2.2524E−02
	200	**0**	**0**	**0**	**0**	9.9413E−01	5.3231E−01	2.1258E+03	1.0187E+02	1.3370E−02	1.9581E−02	1.1966E−02	1.8543E−02
	300	**0**	**0**	**0**	**0**	1.6791E+00	1.2232E+00	3.3015E+03	1.1318E+02	4.2620E−02	4.5935E−02	4.4488E−02	3.5748E−02
	500	**0**	**0**	9.9920E−17	3.3876E−17	3.5160E+00	1.4153E+00	5.6355E+03	1.7792E+02	3.0572E−01	1.4269E−01	3.0168E−01	1.5343E−01
F12 (Generalized penalized function 1)	10	**1.5333E−20**	**4.0214E−20**	1.6929E−19	6.0975E−19	1.1479E+00	8.7994E−01	6.7608E−04	3.3834E−04	1.3023E−03	4.9447E−03	1.3181E−03	5.0044E−03
	20	**9.4594E−10**	**1.4102E−09**	1.0855E−09	1.6338E−09	9.9857E−01	8.4019E−01	4.7009E−04	2.6920E−04	2.3501E−02	1.5776E−02	1.9147E−02	1.9243E−02
	30	**5.2173E−07**	**5.0007E−07**	7.6872E−07	6.0126E−07	1.3678E+00	1.6011E+00	1.5576E−02	4.8859E−02	5.0587E−02	2.9948E−02	7.8106E−02	1.0446E−01
	60	5.6576E−03	4.1137E−03	5.6741E−03	**3.3981E−03**	7.1946E+00	7.4362E+00	**2.7707E−03**	1.2547E−02	2.3356E−01	1.0058E−01	2.5061E−01	7.4331E−02
	100	0.0345	**0.0082**	0.414	0.0118	2.5435E+01	2.8487E+01	**3.7504E−03**	**1.4142E−02**	4.7596E−01	1.1612E−01	4.7849E−01	1.2550E−01
	150	1.1323E−01	2.9546E−02	1.0973E−01	1.9684E−02	5.4380E+05	1.6721E+06	**2.2329E−03**	**3.3197E−03**	8.8172E−01	6.5411E−01	8.7087E−01	4.0700E−01
	200	2.0845E−01	3.1078E−02	2.0568E−01	2.6382E−02	4.4398E+05	1.2292E+06	**2.7505E−03**	**6.2477E−03**	1.6132E+00	1.0729E+00	1.3642E+00	6.8488E−01
	300	3.2934E−01	3.3608E−02	3.5403E−01	2.8132E−02	1.5890E+06	2.9230E+06	**4.9705E−03**	**6.2435E−03**	5.3001E+00	3.6876E+00	4.5466E+00	5.1058E+00
	500	0.6438	0.0348	0.6966	0.0254	1.5884E+07	2.0471E+07	**4.2476E−03**	**5.6032E−03**	8.4733E+05	3.9811E+06	1.0890E+06	4.7425E+06
F13 (Generalized penalized function 2)	10	**1.8336E−18**	**4.7671E−18**	2.3399E−18	6.6144E−18	8.1479E−01	1.6373E−01	1.9961E−03	2.3289E−03	9.9630E−03	3.0353E−02	1.5082E−02	5.7245E−02
	20	1.0204E−02	2.6886E−02	5.0771E−03	1.9262E−02	1.8125E+00	2.9953E−01	**6.0256E−03**	**5.0184E−03**	2.7547E−01	1.4123E−01	3.2363E−01	1.5141E−01
	30	0.0353	0.0529	0.0378	0.0543	3.3179E+00	6.2698E−01	**1.1934E−02**	**1.9387E−02**	1.0527E+00	3.0452E−01	1.0001E+00	2.1791E−01
	60	1.2786E+00	3.6773E−01	1.3418E+00	3.7396E−01	2.7084E+01	2.6343E+01	**1.5791E−02**	**5.2098E−03**	3.6793E+00	3.5972E−01	3.7585E+00	3.8329E−01
	100	5.9526	0.9817	5.9753	1.3649	1.1895E+03	3.9019E+03	**4.9238E−02**	**7.6457E−02**	7.6095E+00	5.1261E−01	7.7334E+00	4.0197E−01
	150	1.2772E+01	7.6514E−01	1.3075E+01	1.0233E+00	1.3366E+04	4.0598E+04	**4.6150E−02**	**7.2167E−02**	1.3263E+01	7.8934E−01	1.3305E+01	7.4707E−01
	200	1.7976E+01	5.7913E−01	1.8853E+01	4.6841E−01	2.0937E+06	6.9315E+06	**1.6624E−01**	**2.9616E−01**	2.0893E+01	2.6301E+00	2.0961E+01	2.9722E+00
	300	2.8771E+01	3.5795E−01	2.9316E+01	2.9528E−01	4.6767E+06	7.2122E+06	**1.1452E−01**	**1.4907E−01**	5.1852E+01	2.6357E+01	5.1780E+01	2.5799E+01
	500	49.5841	0.1305	49.1521	0.2423	1.7675E+07	2.6298E+07	**1.8839E−01**	**2.2036E−01**	5.5538E+04	1.4970E+05	6.4903E+04	2.1823E+05
Friedman mean rank		**1.4815**		2.0926		5.1111		4.0556		4.0833		4.1759	
Rank		**1**		2		6		3		4		5	

Table 5
Statistical results and comparison of the multimodal benchmark functions with the fixed dimensions.

Function	AEO		EO		GWO		WDO		SSA		WOA	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F14 (Shekel's foxholes function, $d = 2$)	**0.9980**	**1.4867E−16**	**0.9980**	1.5412E−16	10.4631	5.4391	3.5813	2.2029	1.0641	0.3622	**0.9980**	1.7208E−12
F15 (Kowalik's function, $d = 4$)	0.0011	0.0037	0.0037	0.0076	0.0079	0.0092	**5.0138E−04**	**1.9269E−04**	7.0650E−04	2.4212E−04	8.4577E−04	3.0028E−04
F16 (Six-hump Camelback function, $d = 2$)	**−1.0316**	**5.9752E−16**	**−1.0316**	6.4539E−16	−1.0205	1.5582E−02	**−1.0314**	2.2288E−04	**−1.0316**	1.3389E−10	**−1.0316**	1.0951E−10
F17 (Branin function, $d = 2$)	**0.3979**	**0**	**0.3979**	**0**	0.3995	6.4766E−03	0.4711	2.0973E−01	**0.3979**	1.0605E−08	**0.3979**	3.1440E−08
F18 (Goldman-Price function, $d = 2$)	**3**	**1.3550E−15**	**3**	1.6759E−15	6.6159	9.3293E+00	3.9184	4.9291E+00	**3**	2.9677E−08	**3**	2.4445E−08
F19 (Hartman's family, $d = 3$)	**−3.8628**	**2.4546E−15**	−3.8625	2.6102E−15	−3.8527	1.7636E−02	−3.8524	7.4392E−03	−3.8596	3.9271E−03	−3.8607	3.5449E−03
F20 (Hartman's family, $d = 6$)	**−3.2841**	**0.0599**	−3.2785	0.0659	−3.1686	0.1740	−3.1142	0.1257	−3.2148	0.1163	−3.1643	0.1559
F21 (Sheke's family, $d = 4$, local minima $m = 5$)	−8.4539	**2.4443**	**−8.4654**	2.6807	−5.8907	3.4945	−4.6822	2.8030	−6.0382	3.1240	−4.9064	2.9191
F22 (Sheke's family, $d = 4$, local minima $m = 7$)	**−9.4716**	**2.1302**	−9.1716	2.5380	−7.4809	3.5331	−5.1875	2.7700	−6.5767	3.5388	−7.1987	3.5411
F23 (Sheke's family, $d = 4$, local minima $m = 10$)	**−9.8154**	**1.8698**	−9.3684	2.6898	−6.3692	3.8460	−4.9635	2.8264	−6.9341	3.5272	−7.6599	3.6578
Friedman mean rank	**1.95**		2.35		5.00		5.20		3.25		3.25	
Rank	**1**		2		4		5		3		3	

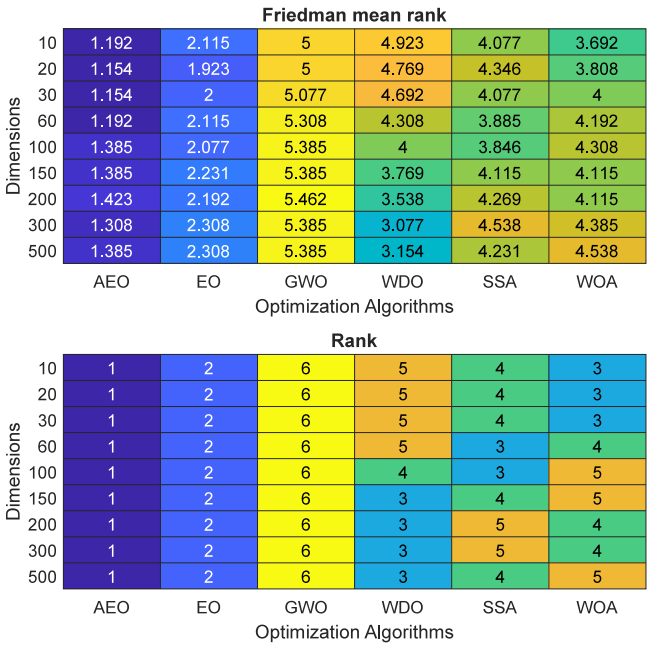

Fig. 8. Friedman's mean rank based on the scalability results from comparisons of scalable unimodal and multimodal benchmark functions F1–F13.

4. The proposed interdependence based multilevel thresholding technique using AEO

The proposed multilevel thresholding method is used to divide the original gray-level distribution into $K+1$ classes using K optimum threshold values. The detailed steps of the proposed method are described below. The flowchart is presented in Fig. 9.

The detail steps of the proposed multilevel thresholding approach are described below:

Step 1: Input the gray level image, the number of thresholds , and d in AEO.
Step 2: Compute the histogram of the input gray level image.
Step 3: Input the histogram and number of thresholds into the AEO to compute the entropic information.
Step 4: Randomly initialize the position of N search agents for d dimension using the Eq. (8).
Step 5: Evaluate and update the optimal threshold values using the AEO, by minimizing the objective function, using the Eq. (7), till the termination criterion is not met.
Step 6: Output the optimal threshold value $\vec{C}_{eq(1)} = \left(t_1^*, t_2^*, \ldots, t_K^*\right)$.

After the optimum threshold values are obtained, the reconstruction of the image (thresholded image) using the threshold values is done.

5. Results and discussions

The experiments are conducted using MATLAB R2018b on Intel core-i3 (6th Gen.) CPU with 8 GB RAM under the Windows 10 environment. For an effective illustration of our proposed algorithm, experiments are performed on 500 images from the Berkeley Segmentation Data set (BSDS500) (Martin et al., 2001). The different threshold levels K (2, 3, and 5) are used for the multilevel thresholding performance evaluation. The test images from the BSDS500 are resized to 256 × 256, and some sample images along with their histograms are presented in Fig. 10. For a quantitative performance evaluation, significant metrics like — the peak signal to noise ratio (PSNR) (Agrawal et al., 2013), the feature similarity index for image (FSIM) (Zhang et al., 2011) and the structural similarity index for image (SSIM) (Zhou et al., 2004) are considered.

Table 6
Statistical results and comparison of AEO, EO, GWO, WDO, SSA, and WOA (computed over 500 images from BSDS 500).

K	Statistical parameter	Nature-inspired optimization algorithm					
		AEO	EO	GWO	WDO	SSA	WOA
2	fit_{ave}	**0.1913**	0.1939	0.2344	0.2669	0.1924	0.1921
	std	**0.0659**	0.0675	0.0662	0.0736	0.0670	0.0668
	$PSNR_{ave}$	**21.6242**	21.5260	21.3882	21.3594	21.3617	21.2602
	$SSIM_{ave}$	**0.6537**	0.6526	0.6483	0.6479	0.6467	0.6446
	$FSIM_{ave}$	**0.6703**	0.6702	0.6633	0.6643	0.6628	0.6619
	Ave_time	0.6736	0.6794	0.6655	**0.6628**	0.6798	0.6635
3	fit_{ave}	**0.3833**	0.3843	0.4922	0.5504	0.3900	0.3846
	std	**0.0967**	0.0978	0.0995	0.1180	0.1010	0.0984
	$PSNR_{ave}$	**23.9297**	23.8908	23.6699	23.6565	23.6445	23.5757
	$SSIM_{ave}$	**0.7283**	0.7267	0.7170	0.7189	0.7164	0.7132
	$FSIM_{ave}$	**0.7424**	0.7408	0.7307	0.7316	0.7298	0.7266
	Ave_time	0.7102	0.7223	0.6869	**0.6871**	0.7337	0.7085
5	fit_{ave}	**0.9659**	0.9661	1.2344	1.6000	0.9900	0.9688
	std	**0.1642**	0.1649	0.2044	0.3865	0.1660	0.1651
	$PSNR_{ave}$	**27.0429**	26.9574	26.5761	25.7083	26.9858	26.6650
	$SSIM_{ave}$	**0.8090**	0.8084	0.8036	0.7838	0.8054	0.8009
	$FSIM_{ave}$	**0.8233**	0.8225	0.8173	0.7997	0.8206	0.8162
	Ave_time	**0.7652**	0.8052	0.7875	0.7813	0.8287	0.7783

To compare the performance of the proposed method, some of the state-of-the-art nature-inspired optimization algorithms (EO, GWO, WDO, SSA, and WOA) are considered. Nevertheless, it is crucial to choose the control parameters to achieve the best results. The optimal configuration and the parameters for AEO, EO, GWO, WDO, SSA, and WOA are listed in Table 2, with the number of search agents $N = 30$ and the maximum number of iterations $\max\_iter = 100$. Then the same parameters are used for all 500 images of the BSDS 500 dataset, for evaluation. As nature-inspired algorithms have randomized behavior, all the experiments are repeated for 30 times for each image.

The quantitative performance metrics are evaluated using the average fitness value (fit_{ave}), standard deviation among the average fitness values (std), the average PSNR ($PSNR_{ave}$), the average SSIM ($SSIM_{ave}$), the average FSIM ($FSIM_{ave}$) and the average time is taken (Ave_time) for an image. The quantitative performance comparisons are presented in Table 6. As the method is a minimization problem, the minimum value of fit_{ave} shows a better result. The performance of the AEO has shown a greater improvement over the GWO and DWO. Noteworthy differences are seen over the EO, SSA, and WOA, resulting in explicit improvements. The performance of the AEO in $K = 2$ and 3 has significant improvement over the other algorithms. However, it gives quite impressive results at $K = 5$ while comparing the average value of PSNR, SSIM, and FSIM results. Considering the average time taken to get the optimal threshold value at the threshold level $K = 2$ and 3, the WDO is quicker than the other. Whereas the AEO is quicker than the other methods while considering the higher threshold level $K = 5$.

The sample test images with the subject identification numbers 8068, 35 010, and 105 019 (with three different modalities of the histogram) are taken for the qualitative performance evaluation. Outputs are visually illustrated in Figs. 11–13. The thresholded images corresponding to the subject with identification number 8068 are presented in Fig. 11. The results obtained for the subject 35 010 are presented in Fig. 12. The outputs for the subject 105 019 are presented in Fig. 13. It can be easily visualized that the thresholded images using our method looks more like the original image. The convergence curves of the average fitness so far vs. iteration are shown in Fig. 14. The left side of Fig. 14 shows the convergence curves of the subject 8068 at a threshold

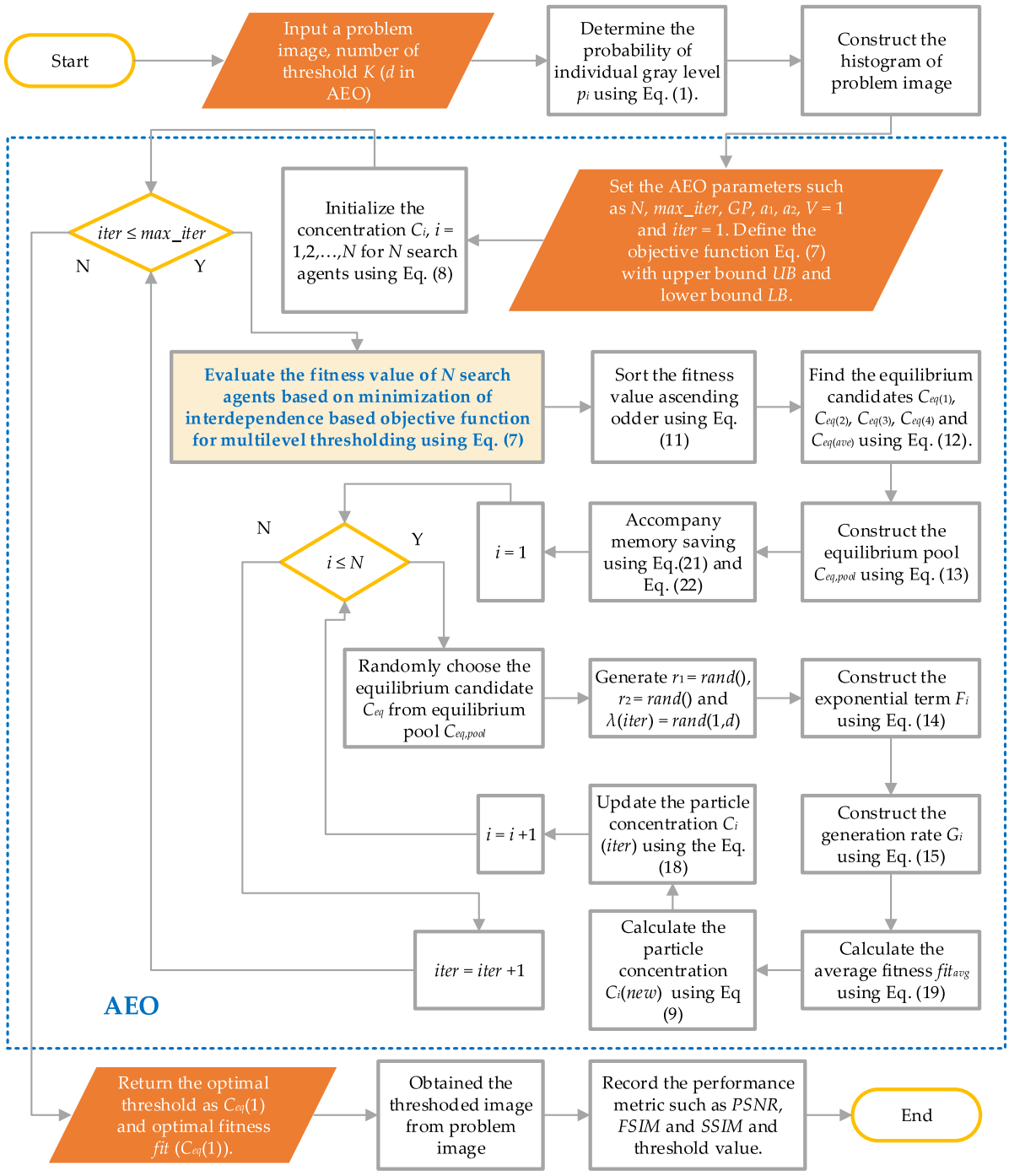

Fig. 9. Flowchart of the suggested interdependence based multilevel thresholding technique using AEO.

level $K = 5$, middle of the Fig. 14 shows the convergence curve of the subject 35 010 at a threshold level $K = 3$, and the right side of the Fig. 14 shows the convergence curves of the subject 105 019 at a threshold level $K = 2$. From Fig. 14, it is observed that the tracking of the optimal fitness value in AEO, EO, SSA, and WOA is much more similar. Hence, it is concluded that the equilibrium optimizer-based algorithm may be used to supplement other optimization algorithms. The proposed multilevel thresholding technique may be explored further to encourage the researchers to work in the field of image processing.

A profound analysis is done in connection with the computation time. To make the method competent, we need to achieve faster convergence. The following Table 7 makes it explicit. The CPU time is compared with four other methods — Otsu's between-class variance, Tsallis entropy, Kapur's entropy, and Masi entropy. Our method has shown noteworthy differences. It enhances the computation speed because the computations involved in calculating the interdependency is less compared to other ideas. For instance, at a threshold level $K = 5$, our method shows about 90% better performance than Otsu's method. However, it has shown about 5% speed improvement while considering the Masi entropy at $K = 5$. It seems that the Masi entropy-based approach is the second contestant. To be precise, the proposed method is much superior to other methods.

An in-depth statistical analysis is provided here for a completeness of the work. It is found from the literature that the statistical results are better analyzed with the help of Wilcoxon's rank test, which is a pair-wise test. This test detects the significant differences between

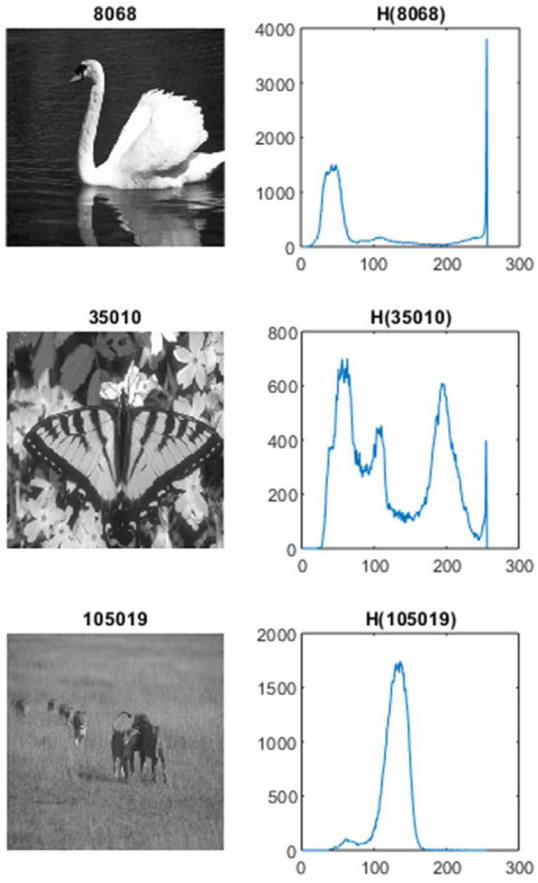

Fig. 10. Sample test image (subject identification number 8068, 35 010, and 105 019) along with their histograms form the BSDS 500 dataset.

the mean value among two pairs of samples. Here, we have taken the average fitness value of 30 independent runs of all 500 images from the BSDS 500 dataset as the samples. The suggested multilevel thresholding method using the AEO algorithm is compared with the other five nature-inspired algorithms. The p-value and the h indicator are used to govern whether accept or reject the null hypothesis. The

Table 7
CPU time comparison of the AEO in a sample image (subject 8068) for different multilevel thresholding techniques.

K	Average time in seconds for ten independent runs				
	Proposed	Otsu	Tsallis	Kapur	Masi
2	**0.6675**	5.3392	0.8944	1.4861	0.7241
3	**0.6921**	6.6445	0.9361	1.5278	0.7569
5	**0.7768**	9.1651	1.0121	1.6038	0.8206

p is the statistical probability, $h = 1$ means the null hypothesis can be rejected at a 5% level of significance, and $h = 0$ means the null hypothesis can be accepted. Based on this, it is seen that the AEO accepts the null hypothesis for EO and WOA algorithm while rejects the null hypothesis in GWO, WDO, and SSA algorithms. From the statistical data from Table 8, the AEO has shown significant improvement over the GWO, WDO, and SSA algorithms. Whereas, as compared to EO and WOA it has shown marginal improvement in multilevel thresholding performance.

6. Conclusion

This paper proposes an adaptive equilibrium optimizer (AEO), an improvement over the equilibrium optimizer (EO), by adapting the adaptive decision making of dispersal of the nonperformer search agents. The AEO shows better performance without increasing the complexity of its predecessors EO. The qualitative and quantitative analyses of the AEO are compared with other optimizers such as EO, GWO, WDO, SSA, and WOA on well-known benchmark functions. Results show the effectiveness of the AEO to obtain the optimal or near-optimal solutions by proper maintaining the exploration and the exploitation abilities. The AEO may be used to solve optimization problems in the field of the artificial intelligence, to obtain the optimal solutions. The AEO would be extended to multi-objective, to tackle the multi-objective problems. The possible extension of the AEO with a crossover–mutation scheme, chaos-based phases, opposition-based learning are optimistic for the future work.

The effectiveness of the AEO in the multilevel thresholding, an important operation of the computer vision, a subfield of the AI, to understand and interpret the data in the real-world are taken for experimentations. Unlike earlier approaches, which are mostly based

Fig. 11. Thresholded images with its corresponding threshold values of the subject 8068 using AEO, EO, GWO, WDO, SSA, and WOA.

Fig. 12. Thresholded images with its corresponding threshold values of the subject 35 010 using AEO, EO, GWO, WDO, SSA, and WOA.

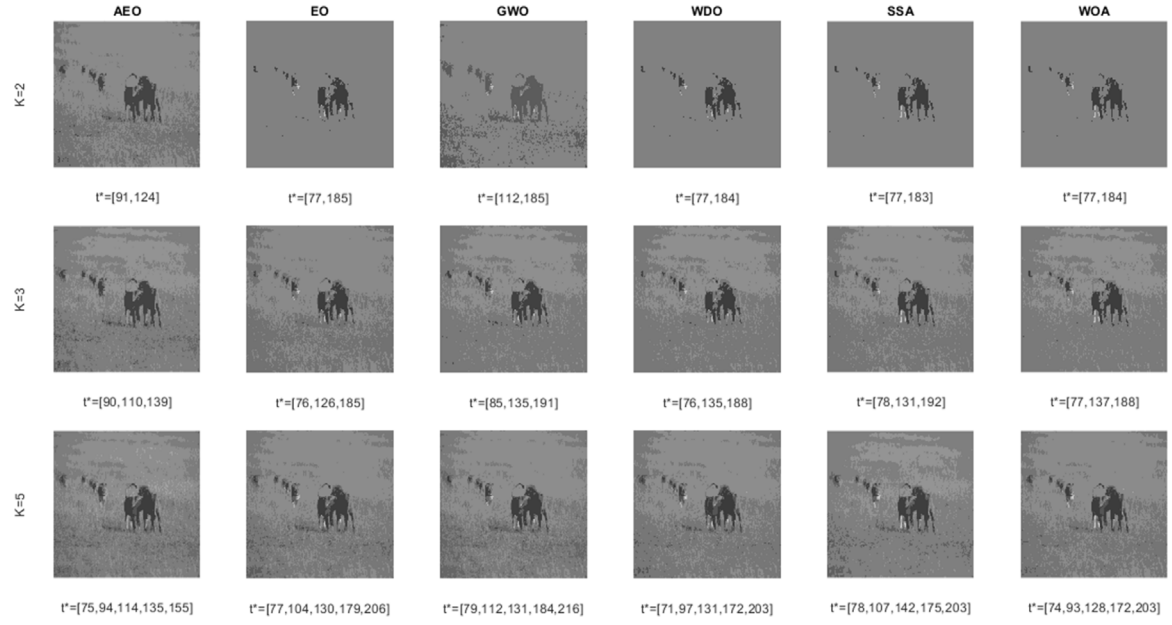

Fig. 13. Thresholded images with its corresponding threshold values of the subject 105 019 using AEO, EO, GWO, WDO, SSA, and WOA.

Table 8
The *p*-value of the Wilcoxon test for 500 images form the BSDS 500 dataset.

K	AEO vs. EO		AEO vs. GWO		AEO vs. WDO		AEO vs. SSA		AEO vs. WOA	
	p	h	*p*	h	*p*	h	*p*	h	*p*	h
2	0.3020	0	3.0150E−22	1	4.1628E−23	1	1.0096E−05	1	0.8923	0
3	0.5052	0	4.1628E−23	1	4.1628E−23	1	1.3988E−12	1	0.3912	0
5	0.7642	0	4.1628E−23	1	4.1628E−23	1	2.9765E−10	1	0.1336	0

on a maximization problem, this work is based on a minimization problem. Remarkable differences are observed in our case on the accuracy plus the execution time because it tries to optimize the shred boundary while using fewer computations to obtain the interdependency between the different classes (regions). The suggested method could extract the parts that have low-contrast inhomogeneous visual features. Therefore, the method is well suited for solving complex segmentation problems. It is reiterated that the method performs better than the existing methods, because of the combined benefits of the impressive accuracy and improved speed. Better performance is achieved, because the shred boundary is optimized by minimizing the interdependency between the different classes. The speed improvement is due to the inherent fewer calculations. A comparison of thresholding results is shown in this paper for convincing the readers. Further, statistical result

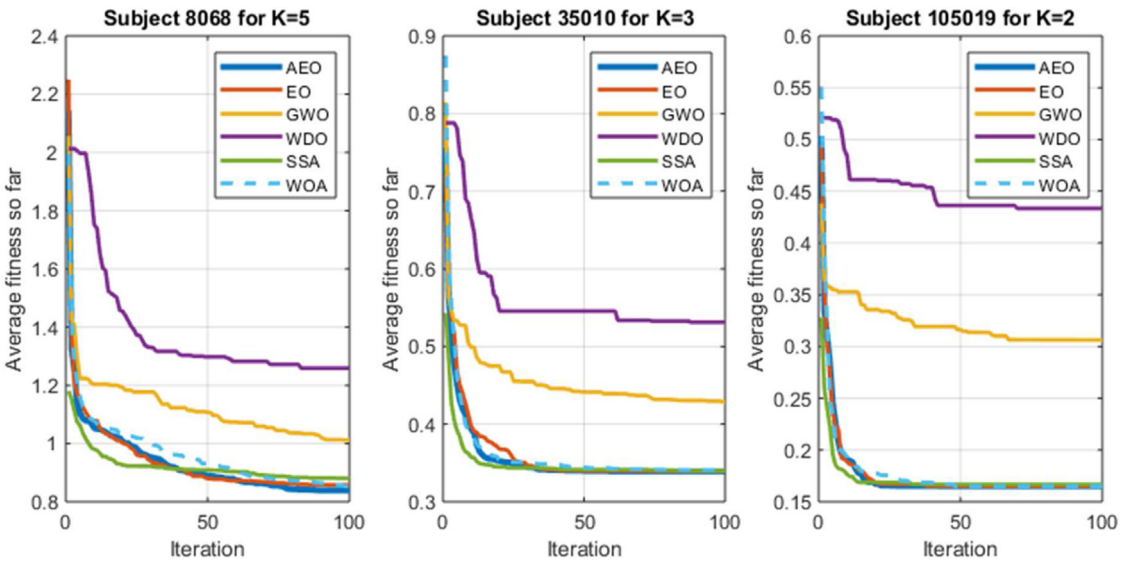

Fig. 14. Convergence curves of AEO, EO, GWO, WDO, SSA, and WOA.

analysis encourages readers for their prospects. The future scope of this work would be the multispectral satellite image analysis, breast cancer thermogram image thresholding and brain magnetic resonance image segmentation for smart healthcare services.

CRediT authorship contribution statement

Aneesh Wunnava: Methodology, Software, Validation, Visualization, Witing - orignal draft. **Manoj Kumar Naik:** Methodology, Validation, Visualization, Writing - review & editing, Conceptualization, Supervision. **Rutuparna Panda:** Methodology, Writing - review & editing, Conceptualization, Supervision. **Bibekananda Jena:** Software, Validation, Visualization. **Ajith Abraham:** Conceptualization, Methodology, Supervision.

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.

References

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.

Agrawal, S., Panda, R., Samantaray, L., Abraham, A., 2017. A novel automated absolute intensity difference based technique for optimal MR brain image thresholding. J. King Saud Univ. - Comput. Inf. Sci. http://dx.doi.org/10.1016/j.jksuci.2017.12.011.

Portes de Albuquerque, M., Esquef, I.A., Gesualdi Mello, A.R., Portes de Albuquerque, M., 2004. Image thresholding using Tsallis entropy. Pattern Recognit. Lett. 25, 1059–1065. http://dx.doi.org/10.1016/j.patrec.2004.03.003.

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

Bayraktar, Z., Komurcu, M., Bossard, J.A., Werner, D.H., 2013. The wind driven Optimization Technique and its application in electromagnetics. IEEE Trans. Antennas Propag. 61, 2745–2757. http://dx.doi.org/10.1109/TAP.2013.2238654.

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.

Castillo, O., Melin, P., Ontiveros, E., Peraza, C., Ochoa, P., Valdez, F., Soria, J., 2019. A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Eng. Appl. Artif. Intell. 85, 666–680. http://dx.doi.org/10.1016/j.engappai.2019.07.020.

El Aziz, M.A., Ewees, A.A., Hassanien, A.E., Mudhsh, M., Xiong, S., 2018. Multi-objective Whale Optimization Algorithm for multilevel thresholding segmentation. In: Hassanien, A.E., Oliva, D.A. (Eds.), Advances in Soft Computing and Machine Learning in Image Processing. Springer International Publishing, Cham, pp. 23–39. http://dx.doi.org/10.1007/978-3-319-63754-9_2.

Elaziz, M.A., Ewees, A.A., Oliva, D., 2020. Hyper-heuristic method for multilevel thresholding image segmentation. Expert Syst. Appl. 146, 113201. http://dx.doi.org/10.1016/j.eswa.2020.113201.

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.

Fu, K.S., Mui, J.K., 1981. A survey on image segmentation. Pattern Recognit. 13, 3–16. http://dx.doi.org/10.1016/0031-3203(81)90028-5.

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.

Johannsen, G., Bille, J., 1982. A threshold selection method using information measures. In: Proc. 6th Int. Conf. Pattern Recogn. Munich, 1982.

Kandhway, P., Bhandari, A.K., 2019. A water cycle Algorithm-based Multilevel Thresholding System for Color Image Segmentation using Masi Entropy. Circuits Syst. Signal Process. 38, 3058–3106. http://dx.doi.org/10.1007/s00034-018-0993-3.

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. Graph. 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.

Khairuzzaman, A.K., Chaudhury, S., 2019. Masi entropy based multilevel thresholding for image segmentation. Multimedia Tools Appl. http://dx.doi.org/10.1007/s11042-019-08117-8.

Küçükuğurlu, B., Gedikli, E., 2020. Symbiotic Organisms Search Algorithm for multilevel thresholding of images. Expert Syst. Appl. 147, 113210. http://dx.doi.org/10.1016/j.eswa.2020.113210.

Liang, J.J., Suganthan, P.N., 2005. Dynamic multi-swarm particle swarm optimizer with local search. In: Evolutionary Computation, 2005. the 2005 IEEE Congress on, Vol. 1. pp. 522–528. http://dx.doi.org/10.1109/CEC.2005.1554727.

Liao, P.-S., Chen, T.-S., Chung, P.-C., 2001. A Fast Algorithm for Multilevel Thresholding. J. Inf. Sci. Eng. 17, 713–727.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 2. pp. 416–423. http://dx.doi.org/10.1109/ICCV.2001.937655.

Martino, F.Di, Sessa, S., 2020. PSO image thresholding on images compressed via fuzzy transforms. Inf. Sci. (Ny) 506, 308–324. http://dx.doi.org/10.1016/j.ins.2019.07.088.

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

Melin, P., Sánchez, D., 2018. Multi-objective optimization for modular granular neural networks applied to pattern recognition. Inf. Sci. (Ny) 460–461, 594–610. http://dx.doi.org/10.1016/j.ins.2017.09.031.

Mirjalili, S., Lewis, A., 2016. The Whale Optimization Algorithm. Adv. Eng. Softw. 95, 51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey Wolf Optimizer. Adv. Eng. Softw. 69, 46–61. http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

Naik, M.K., Panda, R., 2016. A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition. Appl. Soft Comput. 38, 661–675. http://dx.doi.org/10.1016/j.asoc.2015.10.039.

Nie, F., Zhang, P., Li, J., Ding, D., 2017. A novel generalized entropy and its application in image thresholding. Signal Process. 134, 23–34. http://dx.doi.org/10.1016/j.sigpro.2016.11.004.

Otsu, N., 1979. A threshold Selection Method from Gray-Level Histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66. http://dx.doi.org/10.1109/TSMC.1979.4310076.

Pal, N.R., Pal, S.K., 1993. A review on image segmentation techniques. Pattern Recognit. 26, 1277–1294. http://dx.doi.org/10.1016/0031-3203(93)90135-J.

Pavesic, N., Ribaric, S., 2000. Gray level thresholding using the Havrda and Charvat entropy. In: 2000 10th Mediterranean Electrotechnical Conference. Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. MeleCon 2000 (Cat. No.00CH37099), Vol. 2. pp. 631–634. http://dx.doi.org/10.1109/MELCON.2000.880013.

Peng-Yeng, Y., Ling-Hwei, C., 1994. A new method for multilevel thresholding using symmetry and duality of the histogram. In: Proceedings of ICSIPNN '94. International Conference on Speech, Image Processing and Neural Networks, Vol. 1. pp. 45–48. http://dx.doi.org/10.1109/SIPNN.1994.344969.

Rodríguez, L., Castillo, O., García, M., Soria, J., 2020. In: Castillo, O., Melin, P. (Eds.), Constrained Real-Parameter Optimization using the Firefly Algorithm and the Grey Wolf Optimizer BT - Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine. Springer International Publishing, Cham, pp. 155–167. http://dx.doi.org/10.1007/978-3-030-34135-0_11.

Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C.I., Martinez, G.E., Soto, J., 2017. A fuzzy hierarchical operator in the grey wolf optimizer algorithm. Appl. Soft Comput. 57, 315–328. http://dx.doi.org/10.1016/j.asoc.2017.03.048.

Sahoo, P.K., Soltani, S., Wong, A.K.C., 1988. A survey of thresholding techniques. Comput. Vis. Graph. Image Process. 41, 233–260. http://dx.doi.org/10.1016/0734-189X(88)90022-9.

Sankur, B., Sezgin, M., 2001. Image thresholding techniques: A survey over categories. Pattern Recognit. 34, 1573–1583.

Santiago, A., Dorronsoro, B., Nebro, A.J., Durillo, J.J., Castillo, O., Fraire, H.J., 2019. A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME. Inf. Sci. (Ny) 471, 233–251. http://dx.doi.org/10.1016/j.ins.2018.09.005.

Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging 13, 146–168. http://dx.doi.org/10.1117/1.1631315.

Song, J.H., Cong, W., Li, J.J., 2017. A fuzzy C-means clustering algorithm for image segmentation using nonlinear weighted local information. J. Inf. Hiding Multimed. Signal Process. 8, 578–588.

Tsallis, C., 2001. Nonextensive statistical mechanics and its applications. Lecture Notes in Phys. 560, 3–98.

Upadhyay, P., Chhabra, J.K., 2019. Kapur's entropy based optimal multilevel image segmentation using Crow Search Algorithm. Appl. Soft Comput. 105522. http://dx.doi.org/10.1016/j.asoc.2019.105522.

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

Yao, X., Yong, L., Guangming, L., 1999. Evolutionary programming made faster. Evol. Comput. IEEE Trans. 3, 82–102. http://dx.doi.org/10.1109/4235.771163.

Yin, P.-Y., 1999. A fast scheme for optimal thresholding using genetic algorithms. Signal Process. 72, 85–95. http://dx.doi.org/10.1016/S0165-1684(98)00167-4.

Yin, P.-Y., Chen, L.-H., 1997. A fast iterative scheme for multilevel thresholding methods. Signal Process. 60, 305–313. http://dx.doi.org/10.1016/S0165-1684(97)00080-7.

Yue, X., Zhang, H., 2020. Modified hybrid bat algorithm with genetic crossover operation and smart inertia weight for multilevel image segmentation. Appl. Soft Comput. 90, 106157. http://dx.doi.org/10.1016/j.asoc.2020.106157.

Zhang, H., Fritts, J.E., Goldman, S.A., 2008. Image segmentation evaluation: A survey of unsupervised methods. Comput. Vis. Image Underst. 110, 260–280. http://dx.doi.org/10.1016/j.cviu.2007.08.003.

Zhang, L., Zhang, L., Mou, X., Zhang, D., 2011. FSIM: A feature Similarity Index for Image Quality Assessment. IEEE Trans. Image Process. 20, 2378–2386. http://dx.doi.org/10.1109/TIP.2011.2109730.

Zhou, W., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13, 600–612. http://dx.doi.org/10.1109/TIP.2003.819861.