Multimedia Tools and Applications (2022) 81:16707–16742
https://doi.org/10.1007/s11042-022-12001-3

Boosting Marine Predators Algorithm by Salp Swarm Algorithm for Multilevel Thresholding Image Segmentation

Laith Abualigah [1,2] · Nada Khalil Al-Okbi [3] · Mohamed Abd Elaziz [4,5,6,7] · Essam H. Houssein [8]

Received: 26 February 2021 / Revised: 12 July 2021 / Accepted: 4 January 2022 /
Published online: 3 March 2022

Abstract
Pixel rating is considered one of the commonly used critical factors in digital image processing that depends on intensity. It is used to determine the optimal image segmentation threshold. In recent years, the optimum threshold has been selected with great interest due to its many applications. Several methods have been used to find the optimum threshold, including the Otsu and Kapur methods. These methods are appropriate and easy to implement to define a single or bi-level threshold. However, when they are extended to multiple levels, they will cause some problems, such as long time-consuming, the high computational cost, and the needed improvement in their accuracy. To avoid these problems and determine the optimal multilevel image segmentation threshold, we proposed a hybrid Marine Predators Algorithm (MPA) with Salp Swarm Algorithm (SSA) to determine the optimal multilevel threshold image segmentation MPASSA. The obtained solutions of the proposed method are represented using the image histogram. Several standard evaluation measures, such as (the fitness function, time consumer, Peak Signal-to-Noise Ratio, Structural Similarity Index, etc.…) are employed to evaluate the proposed segmentation method's effectiveness. Several benchmark images are used to validate the proposed algorithm's performance (MPASSA). The results showed that the proposed MPASSA got better results than other well-known optimization algorithms published in the literature.

Keywords Image segmentation · Multilevel thresholding · Meta-heuristic algorithms · Marine Predator algorithm · Salp Swarm algorithm

---

✉ Laith Abualigah
Aligah.2020@gmail.com

Extended author information available on the last page of the article

1 Introduction

Image segmentation is an important and challenging process [31]. It is involved in various fields, including pattern recognition, digital image processing, and artificial intelligence [32]. It is considered one of the most critical image processing steps that separate the image components into different parts by merging the same pixel groups. It is also used in the process of extracting object-related features from an image [37]. This is not an easy task; usually, the images contain some unwanted background noise. Various techniques are used in the segmentation process to subtract objects from images and separate or split them from backgrounds [21]. Image segmentation is also an essential step in computer vision. Also, it represents a necessary technique in image analysis, especially for medical image analysis. The images are divided into two types: color and gray ideas. Each of them has different methods of segmentation [46].

Image segmentation methods are classified into layer-based segmentation methods and block-based segmentation methods, as given in Fig. 1. Among the most popular segmentation techniques used are Threshold, Histogram, Edge detection, Watershed Transformation, Clustering, and Region-based methods. As for separating the objects from the background, the Threshold method is used. The segmentation method, which uses threshold, is characterized by its simplicity and accuracy compared to other methods [23]. So, it has received the attention of many researchers in this field.

The threshold method is widely used for dividing pixels in an image into different classes [49, 50]. There are two methods to determine image thresholds: the first is bi-level thresholding, and the second is multilevel thresholding. The first method divides the image into two categories based on the threshold value. It groups pixels densest from a threshold value into one class and pixels less dense than a threshold value into another level. They are often used to separate the foreground and background of the images. The process of dividing into two categories is insufficient, especially when the image is more complex and contains several objects with similar gray levels [45].

Therefore, it is necessary to extend the threshold levels from bi-level to multilevel [44]. The multilevel threshold (MTH) splits an image into several classes belonging to multiple objects

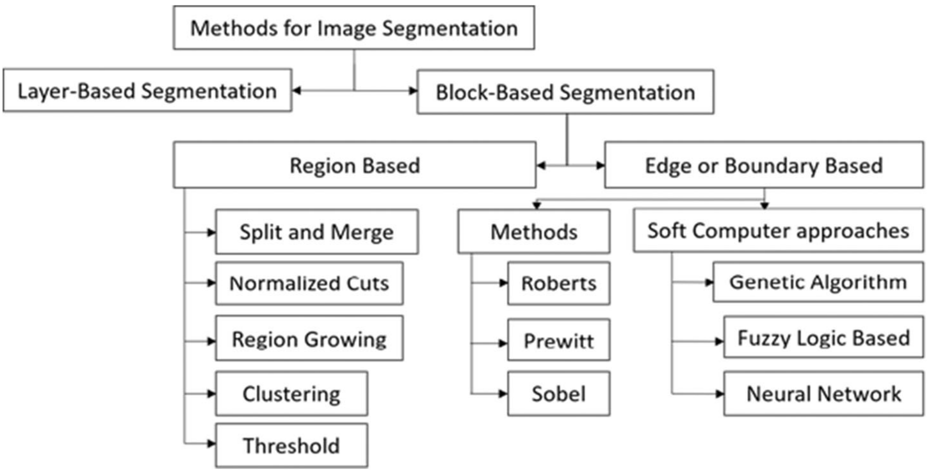

Fig. 1 Methods of image segmentation

in the image. In this method, it is possible to reveal more information and more items from the segmented image. Additionally, several histogram methods have been developed, such as Otsu's method and Kapoor's method. The basis for their work is to maximize the variance between classes and entropy to achieve homogeneity between categories. One of the problems caused by these methods is the complexity of the mathematical operations, primarily as the number of threshold levels increases to obtain better threshold values. Therefore, this problem's solution lies in combining these Metaheuristic algorithms. Several algorithms are used by researchers, such as particle swarm optimization, social-spider optimization [40], and others [13]. In this case, conflict problems arise to achieve a specific objective; this target may not be suitable for image types. This is because each goal has a particular kind of image that makes it powerful.

In the Otsu-based method, the best value of the threshold is maximizing the variance between classes. So, the choice of optimal thresholds in multilevel was a challenge over decades. Which affects segmentation accuracy. To address this weakness, this paper proposes an efficient multilevel threshold method depending on an enhanced Marine Predators Algorithm for image segmentation. The Salp Swarm Algorithm is used as a local search to improve the basic MPA's performance in this proposed method. The proposed algorithm, MPASSA, is a hybrid optimization algorithm for MTH that beats on the shortcomings of individual optimization algorithms using the force of both MPA and SSA. The candidate solutions for the proposed method are represented using an image histogram. The hybrid technique combines the characteristics of two different methods (MPA and SSA). Therefore, the proposed hybrid algorithm (MPASSA) averts getting stuck on a local optimum and has a high ability to find the optimal solution for the image. Several standard evaluation metrics, such as the function of fitness, signal-to-noise ratio, structural similarity index, consumer time, etc., are used to evaluate the proposed segmentation method's effectiveness. The MPASSA is assessed by using two experiments that have a set of images. In the first experiment set, two gray images are used. These images were widely applied in different studies to assess various segmentation algorithms. At the second experiment set, three-color images are used. Also, these images were involved in other studies to evaluate various segmentation algorithms. The results of MPASSA are compared with several optimization algorithms, such as WOA, PSO, AOA, SSA, MPA, and it shows significant performance improvement. The proposed method MPASSA has proved its effectiveness in all test cases. Both experiment sets' performance indicates that the MPASSA is an effective segmentation method, and it can be used in various segmentation applications. The results reveal that the proposed MPASSA is better than other known optimization algorithms published in the literature.

The main contributions of this study are summarized as follows.

1) This research proposes a new image segmentation method using a multilevel threshold based on using optimization method.
2) The SSA operators are used to enhance the exploitation ability of the MPA, called MPASSA.
3) This research tests the proposed method's performance in two experiment sets using two popular grayscale images and three-color popular images.
4) The proposed method is compared with several state-of-the-art methods.

The design of this paper is given as follows. Section 2 contains a literature review of the past and current image segmentation by multilevel thresholding algorithms. Section 3 presents the research methodology and techniques used in Image segmentation by multilevel thresholding algorithms. Section 4 illustrates the implemented proposed algorithms, the evolution measurements that have been used to test and evaluate the proposed algorithms, and research discussion and results. Finally, Section 5 presents the conclusion of this research, followed by suggestions and motivations for future work in the field of Hybrid Algorithms.

2 Background

One of the most critical image analysis problems and processing is the pixel segment into its various categories. The goal of segmentation is to identify elements and isolate them from the background and distinguish between pixels to improve contrast [41]. Several common methods can be used to solve this problem [5, 8, 12, 20, 26, 43, 53]. The most common segmentation techniques that researchers still use nowadays are Threshold, Histogram, Edge detection, Watershed Transformation, clustering, and Region-based methods [1].

There are two types of thresholds; threshold one value is called bi-level thresholds. If more than one threshold value is used, it is called a multilevel point, the second type. In the first type, pixels are divided into two categories. Pixels with an intensity level higher than the threshold value are classified as objects, while the remaining pixels are classified as a part of the background [38]. However, the second one used a multilevel threshold to divide the grayscale images into sections where it became necessary to extend the bi-level threshold levels to the multilevel. When the image is more complex and contains several objects with similar gray levels, it is also used with pictures with colored backgrounds. The multilevel threshold mechanism, which has more than one threshold, divides the image into several classes belonging to multiple objects. Therefore, it results in multiple items with one background. In this way, more information and things can be revealed from the segmented image [2].

Among the most critical methods used with threshold segmentation are the Otsu and Kapur methods [30]. The Otsu method increases the contrast between image classes. In contrast, the Kapur method maximizes entropy as a measure of homogeneity between categories. Both approaches have proven to be one of the most widely used forms of image processing. They are also considered accurate and effective alternatives to pixilation in the bi-level threshold technique. It can be used in a multilevel threshold, but its accuracy decreases as the thresholds' number increases, leading to increased complexity. Many other approaches adopt entropy, such as Tallies entropy, Fuzz,y entropy, the Shannon entropy, and Renyi's entropy. The same problem of increasing computational complexity appears at a multilevel threshold [38]. Therefore, to solve such issues, Meta-Heuristic algorithms provide highly accurate results in most cases. The Meta-Heuristic algorithms are essential tools for solving optimization problems' intricacies with high accuracy [4].

According to the official data provided by those countries, a new forecasting method is proposed in [19] to forecast the number of people infected with Covid-19 in some countries. The proposal offers an improvement to the ANFIS model to predict the number of injured people, depending on another optimizer, the Marine Predator algorithm. The algorithm is used to improve ANFIS parameters, as well as enhance prediction performance. The results of the prediction performance evaluation of the proposed method MPA-ANFIS were compared with several other ways. The result was that the proposed approach significantly outperforms almost all techniques and measures of performance. Among the performance measures that were compared are, mean absolute error (MAE), Root Mean Squared Relative Error (RMSRE), Mean Absolute Percentage Error (MAPE), and others [9, 14].

In [25], a new variant of MPA is proposed. The proposed method's performance is tested on several problems, including the real-world engineering design and CEC-2017 tests. The algorithm was compared to other optimization algorithms: the first is GA and PSO, which is one of the algorithms that have been well studied, the second is GSA, CS, and SSA, as the algorithms developed in the recent period, third is CMA-ES, SHADE, and LSHADE -cnEpSin, are the algorithms that are as optimizers for performance. They are also IEEE CEC competition winners. The results were that the Marine Predator algorithm ranked second as the best performing method, and it showed competitive results compared to LSHADE-cnEpSin. The Marine Predator algorithm is one of the winning algorithms in the CEC 2017 competition. The statistical analysis presented in that paper shown that the Marine Predator algorithm can work as a high-performance optimizer. It also proved that the Marine Predator algorithm is significantly superior to the SSA, GSA, CMA-ES, PSO, CS, and GA algorithms. The study also showed the similarity of performance statistically for each Marine Predator, SHADE, and LSHADE-cnEpSin.

In [34], a novel hybrid algorithm is presented. The Proposed method ensembles Salp Swarm and Multi-objective Salp Swarm in solving the optimization problems for both single and multiple objectives. The Salp's primary behavior is a crowd of strains while on the move to search for food [6]. The researcher tested both algorithms on several mathematical optimization tasks to observe, monitor, and confirm their behavior in finding the best solution. The results showed that the Salp swarm algorithm could optimize the initial random solutions and are close to being the optimum. The results also showed that the multi-objective Salp Swarm algorithm could converge Pareto optimal solutions. The study also showed that both proposed algorithms could solve complex and computationally costly engineering design problems (such as designing marine propellers). The research also indicates that the proposed algorithms have many advantages that distinguish them in solving many real-world situations.

In [3], an effective multilevel threshold (MTH) method is proposed for image segmentation, including medical image segmentation including COVID-19 CT images. Suggested is the Marine Predatory Animal (MPA) algorithm. MPA is a new SI method. According to the researchers' study, the proposal provides the first application for use in image segmentation. The (MFO) algorithm was used in the MPA optimization process. The new proposal has been called MPAMFO. The request was evaluated with various images, as it included cross-sectional images of Coronavirus (COVID-19). The results showed stable and good performance in all tests. Also, several comparisons were applied to prove the MPAMFO proposal's superiority over many algorithms, including PSO, GWO, and CS, in terms of SSIM, fitness value, and PSNR. The proposed algorithm has proven to be highly effective. Therefore, it is possible to improve it and apply it in various improvement processes, including data clustering, machine job scheduling, time series prediction, cloud computing, etc.

A new proposal combining the SSA algorithm with the PSO algorithm is given in [28]. This proposal is to enhance SSA's ability for exploration and exploitation by using PSO characteristics to improve the quality of the SSA for searching for results. Thus, the rate of convergence increases. The proposed algorithm was evaluated with two experiments; first, it was tested on 15 standard groups. The second was applied to determine the optimal subset of the features from among ten UCI groups to increase classification accuracy. The results from the two experiments were compared to several algorithms, including SSA. The work SSA with the PSO algorithm gives better results in performance measures, including chosen feature ratio, processing unit time, accuracy, and evaluating the fitness function. This means that hybridization of the SSA gives better results than it does separately.

In [47], a novel objective function and a new application of the MPA approach are presented to properly extract the nine parameters of the PV module's TDPV model. This study aims to obtain an accurate PV model for any commercial PV panel, which plays an essential role in the grid-connected PV power systems' simulation studies. In this study, the optimization problem's main objective is to minimize a function representing the current error. The MPA technology was successfully utilized to reduce the objective function, obtaining the PV module's TDPV model's nine parameters. Various comparisons were exhibited to check the efficacy of the offered TDPV model using the MPA technology. The proposed algorithm was successfully employed to optimally design the parameters of two marketable KC200GT and MSX-60 PV modules. The optimal parameters, realized using the MPA-TDPV model, are coherent with those achieved using other algorithms. The MPA approach has recorded lower optimal fitness values of 1.245e-14 and 7.458e-13 for KC200GT and MSX-60 PV modules. Furthermore, the simulation outcomes of the MPA-TDPV model concur with the measured data for these well-known PV panels under several environmental situations. The MPA-TDPV model's ACE indicates a lower value than other PV models for the marketable PV panels. This points out the proposed approach's superiority, efficacy, and robustness for achieving a precise TDPV model-based PV panel. With the help of the MPA approach, accurate modeling of any marketable PV panel can be realized. Moreover, the MPA technology can be further extended to solve various optimization problems in power system applications, energy storage devices, and smart grids.

In [52], an innovative objective function with a robust and reliable optimization algorithm named the marine predators' algorithm (MPA) is proposed to provide the optimal pattern structure for three dimensions of PV arrays nine × 9, 16 × 16, and 25 × 25. The MPA is tested with several shade patterns and compared with manta ray foraging optimization (MRFO), Harris hawk optimizer (HHO), and particle swarm optimizer (PSO), as well as the total-cross, tied (TCT) connection. Several quality and statistical measures are computed, such as mismatch power loss, fill factor, percentage power loss, and Wilcoxon signed-rank test to assess the performance of the proposed approach. The research showed that the I-V and P-V characteristics were used to investigate the proposed MPA's applicability compared with the other counterparts. Moreover, the mean execution time has been evaluated. The results reveal that MPA enhanced the PV array power by the percentage of 28.6%, 2.7%, and 5.7% in cases of 9 × 9, 16 × 16, and 25 × 25 PV arrays, respectively, and a uni-peak PV characterizes is achieved as well with lowest execution time and highest consistency in the results across the number of independent runs. Therefore, the authors recommend MPA as an efficient and applicable algorithm for PV reconfiguration systems at any dimension of PV array structures.

In [29], a node localization algorithm has been proposed based on Salp Swarm Algorithm (SSA), which handled the node localization problem as an optimization problem. The proposed algorithm has been implemented and validated in various WSN deployments using other target nodes and anchor nodes. Moreover, the proposed algorithm has been evaluated compared to four well-known optimization algorithms, namely PSO, BOA, FA, and GWO, in terms of localization accuracy, computing time, and several localized nodes. +e obtained simulation results have proved the proposed algorithm's superiority compared to the other localization algorithms regarding the various performance metrics. The proposed approach can be hybridized with different algorithms to reduce the localization error in future work.

In [27], an improved version of the salp swarm algorithm is introduced for predicting chemical compound activities. A set of assessment indicators are used to evaluate and compared with different algorithms, including particle swarm optimization (PSO), Grasshopper Optimization Algorithm (GOA), Grey Wolf Optimizer (GWO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA) using three initialization method, and a superior accuracy was obtained with our proposed approach. Also, compared with other algorithms that used the same data, this research's system has a higher performance using fewer features. The previous algorithms (GOA, GWO, PSO, SSA, SCA, and WOA) are compared. Three different methods were used to initialize the various optimization algorithms to ensure the other optimizers' capability to converge from various initial positions, namely mixed initialization, short initialization, and extensive initialization.

In [10], a new hybrid Arithmetic Optimization Algorithm (AOA) and Deferential Evolution (DE) algorithm (DAOA) is proposed for multi-thresholding image segmentation. The proposed algorithm employs the AOA algorithm to optimize the threshold and then uses this thresholding value to partition the images through DE. So, the DAOA integrates AOA global optimization and DE fast convergence. The experiment results were compared against four algorithms. The DAOA achieved better results than other methods; also, the DAOA provided a faster convergence with relatively lower CPU time. In the future, the DAOA can be applied to other complex image segmentation problems.

We concluded that the given methods in the literature are employed to solve the image segmentation problems by finding the optimal number of threshold values. However, finding these values is a complicated problem and needs an efficient method to determine them [48, 17]. Several optimization techniques proved their ability to solve this proposed and demonstrated that an efficient method is required. Thus, we proposed an effective method to address this problem and finding the optimal threshold values.

3 The research methodology

Before presenting the proposed hybrid algorithm (MPASS), we need to define the problem and understand the mathematical model for the main algorithms: Marine Predators Algorithm and Salp Swarm Algorithm. In this section, we present the formulation of the problem, MPA and SSA, and their main items.

3.1 Formulation of the problem

In this section, the problem formulation of multilevel thresholding is illustrated, which gives a mathematical definition through considering a gray level image *I*. The image *I* is tested to be segmented consisting $K+1$ classes. Each segmentation process requires division pixels of the image *I* into subregions or classes as in Eq. (1). This can be done by determining *K* thresholds $\{t_1, t_2, \ldots, t_K\}$ [22].

$$G_0 = \left\{0 \le I_{ij} \le t_1 - 1\right\}_1 = \left\{t_1 \le I_{ij} \le t_2 - 1\right\}_k = \left\{t_k \le I_{ij} \le L - 1\right\} \quad (1)$$

In given Equations, G_K represents *K-th* class of image *I*, *I (i, j)* is the value of gray level to the pixel *(i, j)*, t_K where *(k = 1,…, K)* that define *k-th* threshold value, and L represents the gray levels of *I*, all levels within the range (*0, 1…L - 1*).

Multilevel thresholding aimed to locate the optimal threshold values that divide *I* into several groups; Optimization methods can determine these values. One of the standard methods and well-known optimization function is Otsu's method [39]. It is used in bi-level segmentation problems as well as in multilevel. In multilevel thresholding, Otsu's technique locates the optimum threshold values of *I* by maximizing as in Eq. (2). It can be defined as the following:

$$t_1^*, t_2^*, , t_k^* = F(t_1, t_2, ., t_k) \tag{2}$$

where $F\ (t_1, ..., t_K)$ is the intra class difference (Otsu's function) that defined in Eq. (3).

$$F = \sum\nolimits_{i=0}^{k} A_i (\eta_1 - \eta_2)^2 \tag{3}$$

$$A_i = \sum\nolimits_{j=t_i}^{t_{i+1}-1} P_j \tag{4}$$

$$\eta_i = \sum\nolimits_{j=t_i}^{t_{i+1}-1} i \frac{P_j}{A_j}, where P_i = h(i)/N \tag{5}$$

where η_1 pointed to mean in tensity of image *I*, with $t_0 = 0$ and $t_{K+1} = L$. $h(i)$ is frequency and P_j is probability of the *ith* gray level. *N* is the all pixels in the evaluated *I*.

3.2 Marine Predators Algorithm (MPA)

This part will explain the Marine Predators Optimization Algorithm (MPA) development process as an efficient and straightforward meta-heuristic optimization method [19]. The basis of the algorithm's work is population. The first solution is distributed randomly on the search area in a uniform manner, as in Eq. (6):

$$X_0 = X_{min} + rand\ (X_{max} - X_{min}) \tag{6}$$

where X_{min} and X_{max} are the minimum and maximum bound for variables, the rand is a random vector within range (*0, 1*).

Based on the theory of survival of the fittest, an array can be constructed to include the best solution for the best predator. The best predator is the one that has the most talent for foraging. This matrix is called the elite. The array is represented mathematically as in Eq. (7).

$$\text{Elite} = \begin{bmatrix} X_{1,1}^{I} & X_{1,2}^{I} & \cdots & X_{1,d}^{I} \\ X_{2,1}^{I} & X_{2,2}^{I} & & X_{2,d}^{I} \\ \vdots & \ddots & \ddots & \vdots \\ X_{n,1}^{I} & X_{n,2}^{I} & & X_{n,d}^{I} \end{bmatrix} \tag{7}$$

As for the prey, it is arranged as in the elite matrix. As the predator sites update depending on this matrix. This means preparing the primary prey on which the (fittest) predator depends in the elite ranking. The prey matrix in Eq. (8).

$$\text{Prey} = \begin{bmatrix} X^I_{1,1} & X^I_{1,2} & \cdots & X^I_{1,d} \\ X^I_{2,1} & X^I_{2,2} & & X^I_{2,d} \\ \vdots & \ddots & \ddots & \vdots \\ X^I_{n,1} & X^I_{n,2} & & X^I_{n,d} \end{bmatrix} \tag{8}$$

In Eq. (8), the *Xi,j* represents the *jth* dimension of (*ith*) prey. It is noted that the optimization process is entirely linked directly and mainly to these two matrices.

The process of optimizing the algorithm and moving it from one phase to another depends heavily on the predator's speed relative to the prey. Therefore, the process of improving the Marine Predators algorithm can be divided into three primary stages, depending on the different speed ratios for both the prey and the predator and the life of each of them. The steps are as follows: 1- The movement of the prey relative to the predator is faster with a high rate. 2- The movement of both predator and prey at equal or close speed. 3- The movement of the predator to the prey is faster, with a low rate.

Phase No. 1 The exploration stage is where the prey moves at high speed to discover new search areas. At this stage, the predator should remain and monitor the prey's movements. This stage's performance is in the first third of the total iterations of development (i.e., **1/3 * M***ax_iter*) as in Eq. (9).

$$stepsize_i \rightarrow = \; R \rightarrow_B \otimes (\, Elitel \rightarrow_i - \; R \rightarrow_B \; \otimes Prey \rightarrow_i)$$

Where, *i*= 1…….*n*

$$Prey_i \rightarrow = \; Prey_i \rightarrow + P \,.\; CF \rightarrow \otimes \; stepsize_i \rightarrow$$

$$CF = (1 - \frac{Iter}{Max_{Iter}})^{(2\frac{Iter}{Max_{Iter}})}$$

Where CF is an adaptive variable for controlling the step volume of a predator. R_B represents random numbers and is represented by a vector. These numbers are regular in *[0,1]* and represent *Brownian motion*. *P* = 0.5 represents a vector and constant number for random numbers. The variable *Iter* represents the current iteration, while *Max_iter* represents the maximum iteration.

Phase No. 2 At this point, both the predator and the prey are moving at the same speed. This part occurs in the middle stage of improvement in Eq. (10). Depending on the rule, the unit pace ratio *(v ≈ 1)*, if prey moves in Lévy, the best-organized predator is Brownian motion. This can be considered a transitional stage between exploration and exploitation. Both exploitation and exploration are essential in this section.

$$While\ 1/3\ Max\_Iter < Iter < 2/3\ Max\_Iter$$

$$stepsize_i \rightarrow = \ R \rightarrow_L \otimes (Elitel \rightarrow_i - \ R \rightarrow_L \otimes \ Prey \rightarrow_i)$$

$$Prey_i \rightarrow = \ Prey_i \rightarrow +P\ .\ CF \rightarrow \otimes stepsize_i \rightarrow$$

Where R_L refers to the random numbers, where these numbers are following the Lévy movement in distribution, the first half represents exploitation (represented by predators), Eq. (11) is applied to it. While the prey that represents the second half of the population assumes:

$$stepsize_i \rightarrow = \ R \rightarrow_B \otimes (R \rightarrow_B \otimes \ Elitel \rightarrow_i - \ Prey \rightarrow_i)$$

$$Prey_i \rightarrow = \ Elitel_i \rightarrow +P\ .\ CF \otimes stepsize_i \rightarrow$$

Max_iter is the overall number of generations. The R_B and *Elite* simulate the predator's movement in a Brownian motion, while the *prey* changes its position depending on the predator movement in Brownian fashion.

Phase No. 3 This phase is the last of the optimization process. They represented when the predator's movement is faster than a prey. The predator exploits the prey that is detected and attack it very quickly to get it. So, the exploitability is often high. The best motion for the predator in a low pace ratio *(v = 0.1)* is *Lévy*. This phase is executed on the last third of the iteration numbers *(Iter > 2/3 Max_iter)*, the predator follows *Lévy*. The predator will update its position in Eq. (12).

$$stepsize_i \rightarrow = \ R \rightarrow_L \otimes (R \rightarrow_L \otimes \ Elitel \rightarrow_i - \ Prey \rightarrow_i)$$

where,

$$Prey_i \rightarrow = \ Elitel_i \rightarrow +P\ .\ CF \otimes stepsize_i \rightarrow$$

Step size is added to the *Elite* site for the predator's movement to update the prey site. The doubling of the R_L and *Elite* also simulates the predator's movement within Lévy's strategy.

3.2.1 Eddy formation and FADs' Effect

Another point that causes a behavioral change in marine predators is environmental issues such as the eddy formation or Fish Aggregating Devices (FADs) effects. As mentioned in [28], Sharks spend over 80% of their time in the vicinity of *FADs*. The others are taking the long jumps in different dimensions to find other prey. The effects of *FADs* can be mathematically expressed in Eq. (14).

$$\overrightarrow{Prey_i} = \begin{cases} \overrightarrow{Prey_i} + CF\left[\overrightarrow{X}_{min} + \overrightarrow{R}_i\right] \otimes \overrightarrow{U} & if\ r \le FADs \\ \overrightarrow{Prey_i} + [FADs(1-r)+r]\ \overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}} & if\ r > FADs \end{cases}$$

Where *FADs = 0.2* represent the probability of affected *FADs* on the process of the optimization. *U* is a vector of binary with matrices that have contained zero and one. This is done by generating an arbitrary vector in *[0,1]* and update the array to zero if it's less than *0.2*, and one if it's more. *r* is the uniform arbitrary number *[0,1]*. Both (X_{min}, X_{max}) are the vectors consist of (*lower, upper*) bounds of the dimensions. r_1 and r_2 represent indexes of random prey.

3.2.2 Marine memory

Figure 2 illustrates the second phase of optimization, in which the predator adopts the Brownian strategy better in searching for its prey within the field, as indicated in blue.

When optimization is in its final phase, algorithms need a high ability for exploitation. The third phase represents the last phase of optimization when the predator changes its behavior from Brown's strategy to Levi so that the search process will be more efficient in a particular area. At the same time, the Convergence Factor (*CF*) has an excellent role for predators. This limits the search in several parts of the specific area for exploitation. It also avoids the effort in the search that extends from long steps as a result of using Levi's strategy for non-promising areas in the field. The flowchart of the MPA algorithm is represented as shown in Algorithm 1.

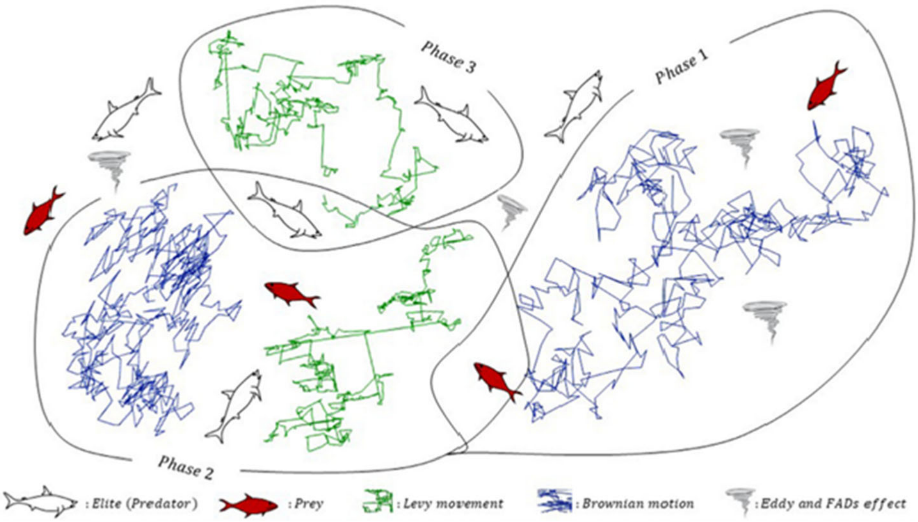

Fig. 2 The three MPA optimization phases

Algorithm 1: The procedure of the Marine Predators Algorithm
Initialize *search agents (Prey) populations i=1,…,n*
While *termination criteria are not met*
Calculate *the fitness and construct the Elite matrix*
If *Iter<Max_Iter/3*
Update *prey based on Equation (6)*
Else if *Max_Iter/3 < Iter < 2 * Max_Iter/3*
For *the first half of the populations (i=1,…,n/2)*
Update *prey based on Equation (7)*
For *the other half of the populations (i=n/2,…,n)*
Update *prey based on Equation (8)*
Else if *Iter>2*Max_Iter/3*
Update *prey based on Equation (9)*
End (if)
Accomplish memory saving and Elite update
Applying FADs effect and update based on Equation (10)
End while

3.3 Salp Swarm Algorithm (SSA)

The Salp algorithm is one of the meta-heuristic algorithms that has been successfully used in solving many optimization problems in various fields [34]. Mathematically, as we know, that the swarm is divided into a leader and followers. The leader guides his followers in their movements. The X represents a swarm of n of the Salps as Eq. (15), represented by a two-dimensional matrix. The fitness of each Salp is calculated to determine the best among them (meaning, defines the leader for the swarm). The leader positions will be updated by using Eq. (16).

$$x_i = \begin{bmatrix} X_1^1 & X_1^1 \cdots & X_{1,d}^1 \\ X_1^2 & X_2^2 & X_d^2 \\ \vdots & \ddots \ddots & \vdots \\ X_1^n & X_d^n & X_d^n \end{bmatrix} \tag{15}$$

$$X_i^1 = \{ yi + ((ub_i - lb_i)r2 + lb_i) \quad r4 \tag{16}$$

*Wherexi*1represents the position for the leader (first) Salp in the dimension *ith*, and y_i is the food site in the *ith* dimension. lb_i and ub_i is the lower and upper bound at the *ith* dimension, respectively, and the coefficient r_1 is calculated by Eq. (17). r_2 and r_3 are random numbers between [0,1].

$$r_1 = e^{-\left(\frac{4l}{L}\right)2} \tag{17}$$

Where L is the upper iterations and l is the running iteration. Note that the coefficient r1 is very significant in SSA because the balances of exploration and exploitation depend on it during the entire search process. As for the followers, Eq. (18) shows the updated positions.

$$x_i^j = \frac{1}{2}\lambda t^2 + \delta_0 t \tag{18}$$

Where $j >= 2, xi1$pointed to the site of the nth Salp in the *ith* dimension, δ_0 represents an initial speed, t is the time, and $\delta_0 = \frac{x - x0}{t}$. The time in optimization points to the iteration. So, the difference between iterations is equal to 1. Considering the proposition that $\delta o = 0$, the following equation (in Eq. (19)) is employed for this issue.

$$x_i^j = \frac{1}{2}\left(x_i^j + x_i^{j-1}\right) \tag{19}$$

Where $j >= 2$. In some cases, the Salp leaving outside of the search space, how we can bring them back to the search space in Eq. (20).

$$x_i^j = \{ l^j \quad if x_i^j \leq l^i j \quad if x_i^j \leq u_i^j j \quad if x_i^j \leq l^i \tag{20}$$

Exploration and exploitation, diversification and intensification, global search and local searches, few pairs of words are common in optimization algorithms. At least the optimization algorithms include one of those pairs. Here we will talk about exploration and exploitation. Exploration aims to discover new areas in the search area. This makes the search process not stop within one level, leading to discovering the optimal solution level. The exploitation aims to reach a better solution within the suitable and discovered solutions. A coefficient responsible for the balance between exploration and exploitation, it is called the coefficient c_1. It is calculated from the equation Eq. (2). The Pseudo-code of the SSA is given in Algorithm 2.

Algorithm 2. The procedure of the Salp Swarm Algorithm
Initialize *the salp population xi (i=1, 2, …., n) considering ub and lb*
While *(end condition is not satisfied) do*
Calculate *the fitness of each search agent (salp)*
F= the best search agent
Update *c1*
For *each salp in (xi) do*
if (i== 1) then
Update *the position of the leading Salp*
Else
Update *the position of the follower salp*
End if
End for
Amend the Salp based on the upper and lower bounds of variables
End while
Return F

1.1 The proposed hybrid algorithm (MPASSA)

The research introduces a hybrid algorithm used to segment images. The proposed hybrid algorithm determines the optimal multilevel threshold values that maximize Otsu's objective function. The proposed algorithm is called MPASSA and is based on the MPA and SSA algorithms and uses Otsu's method as an objective function. The proposed hybrid algorithm depends on boosting the performance of MPA via using the SSA.

The MPA is used to reduce the search region by determining the best solution; then, SSA optimizes each agent less than the limited base. Therefore, the ABCSCA algorithm starts by computing the histogram of an input image and then generates a random population of pop_size solutions (threshold values). Then, the MPA updates these populations using its levels representing by three faces of motion, then Eddy Formation and FADs' Effect. The optimal solution is then determined from the population based on Otsu's method (the best solutions from the MPA algorithm).

The SSA begins determining the minimum threshold value by using the MPA's output (worst solution) and optimizing the solutions of the population via the strategy discussed before. The optimal global solution (the SSA algorithm's output) is determined, and all the previous steps are repeated until the stopping conditions are met. The final stage of the MPASSA is illustrated in Fig. 3. It can be observed that the proposed method introduces an arithmetic complex which is $O\ (t\ (nd\ Cof * n))$, where n represents a number of factors, t is the iteration, d is the problem dimension, and Cof is the cost of function.

4 Experiments and results

In this part, the performance of the MPASSA is evaluated by two experiments (gray and colored). It is compared with five algorithms: original Marine Predators Algorithm MPA, original Salp Swarm Algorithm (SSA), Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), and Arithmetic Optimization Algorithm (AOA). Besides, two types of images are used to test the proposed method (2 grays and 3 color images). The parameters settings of the employed methods are takes from the original methods. The supplied results is expressed as the best fitness function, PSNR, and SSIM. Furthermore, for all of the acquired findings, the Friedman ranking test is used to demonstrate how significantly the suggested approach outperforms the comparable methods for all of the examined issues. The methods are developed and implemented in MATLAB (2015a) on a machine with the following specifications: CPU 2.3 GHz (an Intel Core i7 platform), 16 GB RAM, 2400 MHz, and DDR4.

4.1 Experimental settings

The comparison of all algorithms used in the research has the same cessation conditions (maximum iterations set to 100), with a total run of 30 per algorithm and the population's size (25). For performance evaluation of all experiments on the test images are done by the number of thresholds: 2, 3, 4, and 5 as in [18]. The selection for such thresholds was to illustrate the performance of the proposed algorithm (MPASSA) compared with the traditional

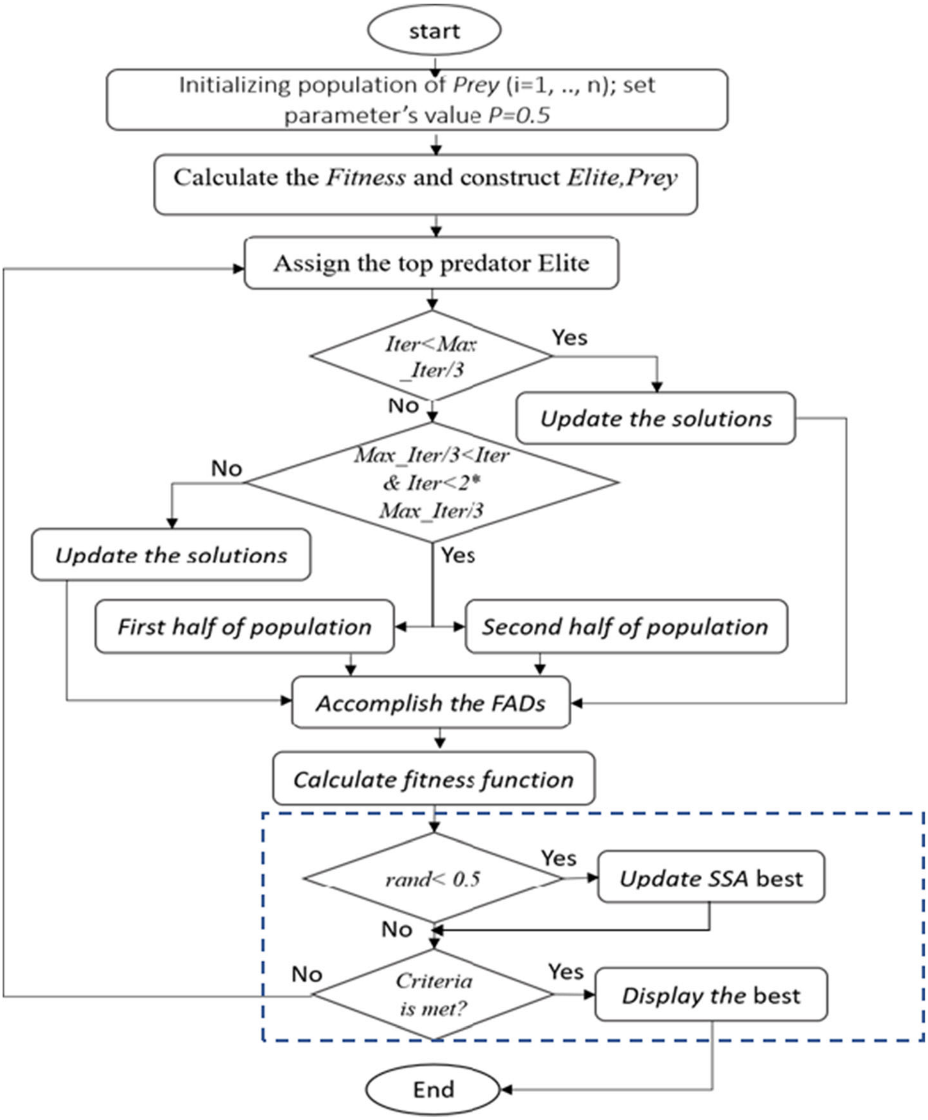

Fig. 3 The proposed method (MPASSA)

segmentation algorithms based on the swarm. All algorithms are programmed in "Matlab2017" and implemented on a "Windows 10 - 64bit" environment on a computer with Intel Corei7 8th Gen (1.99 GHz) processor" 1T" memory and SSD (256).

4.2 Performance Measures

In order to evaluate the fitness of the segmented image, four measures are used. The measures that applied to the proposed algorithms, as follows:

1. The execution time.

2. The fitness value is calculated to evaluate and assess each solution based on its current position, as mention in chapter four.
3. The Peak Signal-to-Noise Ratio (PSNR) measure [51]: used to measure the variance between the reference image and segmented image, and it depends on the value of intensity in the image, and this refers to the fitness of the reconstructed image. The PSNR is defined as:

$$PSNR = 20\left(\frac{255}{RMSE}\right), (in\ dB) \tag{21}$$

RMSE refer to the root-mean-squared error, detect as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}\left(I(i,j) - Seg(i,j)\right)^2}{M*Q}} \tag{22}$$

I and *Seg* refer to the original and segmented images have a size $M \times Q$, respectively. The maximum value of *PSNR* refers to the max performance of the segmentation algorithms.

4. The Structural Similarity Index (SSIM): is used to evaluate the similarity between (*I*) and (*Seg*) image, defined as:

$$SSMI(I, Seg) = \frac{(2\mu_I\mu_{Seg} + C_1)(2\sigma_{I.Seg} + C_2)}{(\mu_I^2 + \mu_{Seg}^2 + C_1)(\sigma_I^1 + \sigma_{Seg}^2 + C_2)} \tag{23}$$

μI and μSeg refer to mean intensity of (*I*) and (*Seg*) respectively; while σI and σSeg refer to standard deviation of (*I*) and (*Seg*) respectively; σI, *Seg* represent a variance of (*I*) and (*Seg*). *c1* and *c2* are constants, that $c_1 = 6.5025$ and $c_2 = 58.52252$ [36]. The maximum value of *SSIM* refers to better performance.

5. Friedman ranking test: The Friedman test is the non-parametric alternative to the one-way ANOVA with repeated measures. It is used to test for differences between groups when the dependent variable being measured is ordinal. It can also be used for continuous data that has violated the assumptions necessary to run the one-way ANOVA with repeated measures [7].

4.3 Parameters setting

Table 1 show the parameters settings for each algorithm that are used in the following experiments. In addition, the general parameters are set as follows.

4.4 Benchmark images

This paper tested all algorithms on the five images: two gray images and three-color images. These images are from the dataset of Berkeley University [33], namely, lena, baboom for gray

Table 1 The parameters of all algorithms and their values

Algorithm	Parameters setting
SSA [34]	c_2 and c_3 are random value [1, 0]
MPA [24]	FAD_S=0.2, P=0.5
PSO [16]	$wMax$=0.9, $wMin$=0.2, c_1=2, c_2 =2
WOA [35]	a [2,0], b=1, t [-1,1]
AOA [42]	$Alpha$=5, Mu=0.499
MPASSA	FAD_S=0.2, P=0.5,

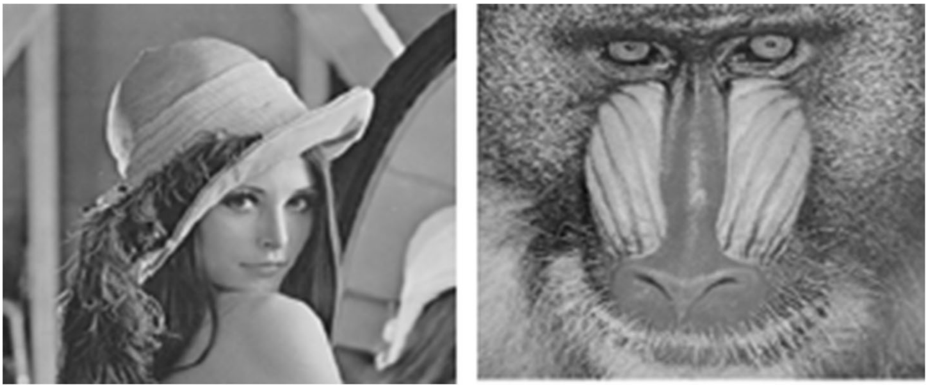

Fig. 4 **a**-Lena_gray, **b**- Baboom_gray

images as shown in Fig. 4(a, b), and lena, baboom, and peppers for color images as shown in Fig. 5(a, b, c). With size (512 × 512) for all.

4.5 The results and discussions

This section discusses the proposed hybrid MPASSA compared to other popular algorithms in the segmentation domain. The comparisons are conducted using several measures (i.e., fitness value, PSNR, SSIM, and time). Two grayscale of images and three-color images with four numbers of thresholds (2, 3, 4, and 5) are applied and compared the results with five comparative algorithms (i.e., MPA [25], SSA [34], WOA [1], PSO [15], and AOA [11])

Fig. 5 **a**-Lena_color, **b**-Baboom_ color, **c**-Peppers_ color

Table 2 The best fitness values obtained from the algorithms on the gray images

Image	K	WOA	SSA	AOA	MPA	PSO	MPASSA
baboom	2	2540.24	2540.321	2539.063	2561.349	2539.553	**2564.773**
	3	2705.651	2705.093	2665.147	3429.821	2720.444	**3441.869**
	4	2761.407	2771.824	2720.958	3811.454	2772.276	**4320.183**
	5	2814.123	2796.225	2775.288	4706.062	2802.736	**4732.528**
lena	2	1961.774	1961.801	1961.048	1970.247	1975.049	**1978.61**
	3	2121.952	2128.221	2070.956	2686.071	2134.601	**2698.594**
	4	2191.709	2186.528	2146.923	2693.825	2320.882	**3217.882**
	5	2194.979	2210.997	2171.321	3555.501	2545.561	**3763.314**

Table 3 The best fitness values obtained from the algorithms on the color images (*Red color*)

Image	K	WOA	SSA	AOA	MPA	PSO	MPASSA
lena	2	1759.116	1758.928	1745.142	1777.281	1775.373	**1779.09**
	3	1885.357	1882.624	1830.816	2400.669	1884.819	**2401.463**
	4	1916.914	1936.248	1851.609	2706.181	1938.726	**2870.553**
	5	2344.27	2344.721	2325.878	4652.723	2418.37	**4752.116**
baboom	2	2903.044	2902.255	2858.549	2916.941	2913.571	**2917.897**
	3	3086.404	3060.594	2981.364	4390.888	3094.46	**4419.153**
	4	3154.283	3153.333	3090.015	4840.728	3156.338	**4973.319**
	5	3200.018	3201.987	3121.989	6248.24	3209.725	**6273.732**
peppers	2	1758.845	1755.602	1705.031	1778.65	1769.214	**1778.743**
	3	1883.002	1847.382	1811.518	2362.56	1881.986	**2405.533**
	4	1923.829	1918.6	1911.359	2538.1	1907.119	**2898.64**
	5	1953.451	1955.783	1895.952	2859.001	1950.474	**3228.724**

The results of the comparison are given in terms of PSNR, SSIM, CPU time, Fitness values, in addition to the Friedman ranking test for the comparative methods using the PSNR and SSIM Mean values for each image. These are standard performance evaluation measures in image segmentation, especially for evaluating multilevel thresholds methods. The discussion depends on each criterion are given as follows.

Table 4 The best fitness values obtained from the algorithms on the color images (*Green color*)

Image	K	WOA	SSA	AOA	MPA	PSO	MPASSA
	2	2430.33	2434.255	2382.197	2467.48	2428.952	**2469.495**
	3	2638.255	2645.489	2495.657	3249.076	2602.186	**3254.056**
	4	2726.173	2727.67	2572.067	3565.544	2746.819	**3636.873**
	5	2746.354	2731.698	2703.694	3896.637	2736.831	**4318.764**
baboom	2	2326.881	2328.222	2251.314	2348.447	2349.319	**2351.324**
	3	2417.967	2417.967	2315.081	3051.351	2432.396	**3244.085**
	4	2472.045	2483.902	2254.491	3146.329	2480.248	**3584.116**
	5	2527.195	2483.655	2433.25	3722.265	2485.464	**4033.038**
peppers	2	5203.927	5203.927	5194.795	5233.783	5227.777	**5235.423**
	3	5381.305	5374.99	5334.478	6916.685	5353.02	**7354.29**
	4	5451.083	5447.217	5418.614	7628.84	5427.893	**9026.392**
	5	5519.578	5509.297	5506.294	8698.647	5506.08	**9259.896**

Table 5 The best fitness values obtained from the algorithms on the color images (*Blue color*)

Image	K	WOA	SSA	AOA	MPA	PSO	MPASSA
lena	2	1009.721	1009.721	727.3778	1033.654	1023.687	**1034.68**
	3	1069.997	1036.101	915.2615	1231.034	1038.308	**1384.995**
	4	1087.979	1008.752	1044.375	1581.942	1057.043	**1594.859**
	5	1103.255	1106.493	1079.437	1659.359	1150.208	**1816.25**
baboom	2	4005.208	4005.208	4003.659	4029.964	4011.061	**4030.464**
	3	4203.425	4203.425	4136.449	4882.566	4218.451	**5927.24**
	4	4292.959	4244.282	4167.025	5844.418	4311.669	**6536.206**
	5	4328.937	4297.179	4291.278	6677.484	4368.803	**7257.704**
peppers	2	1656.159	1656.201	1503.631	1667.147	1669.945	**1672.346**
	3	1765.681	1776.796	1652.439	2085.579	1697.584	**2148.473**
	4	1848.924	1793.298	1762.796	2085.936	1847.228	**2620.117**
	5	1878.877	1879.04	1785.778	2709.747	1851.902	**3019.093**

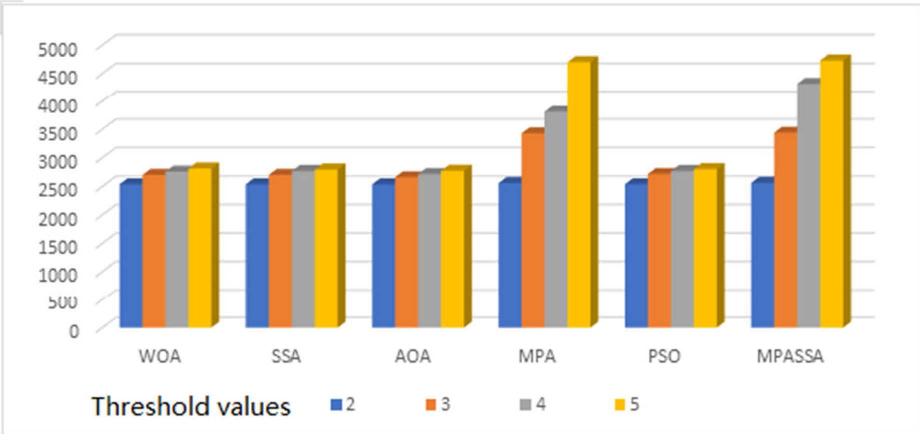

Fig. 6 Fitness values of *baboom* image

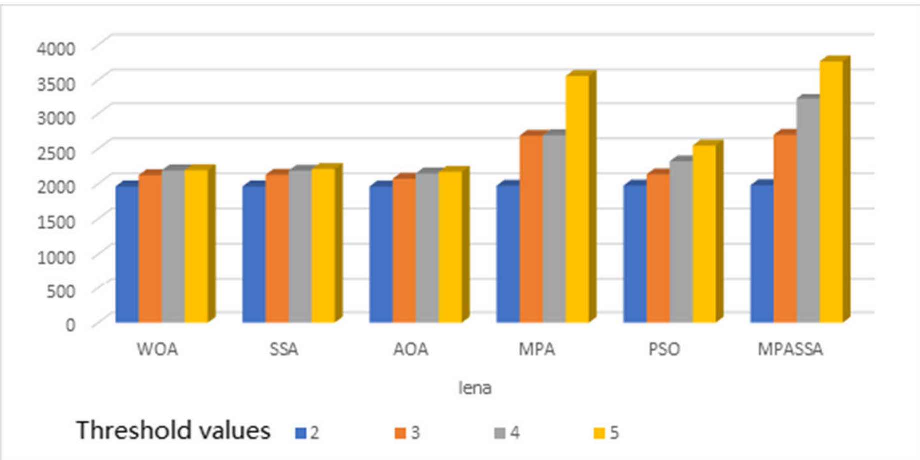

Fig. 7 Fitness values of *lena* image

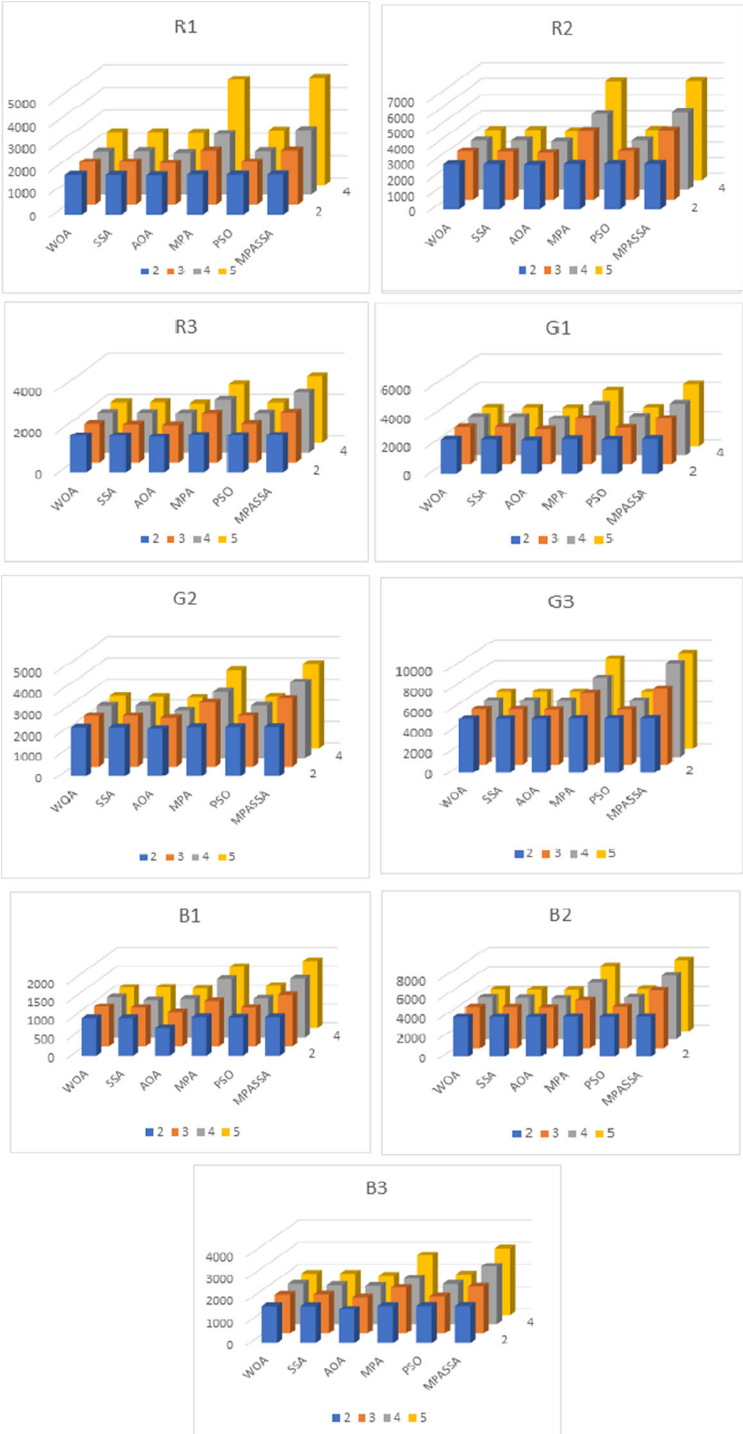

Fig. 8 Fitness values of color images. R is red, G is green, B is blue. And 1 is lena, 2 is baboom, 3 is peppers images

Table 6 Comparison of *PSNR* for gray images

Image	K	Measure	MPASSA	WOA	SSA	AOA	PSO	MPA
lena	2	MAX	15.57891	13.70308	13.90694	15.30526	15.27346	13.19764
		MEAN	14.99636	13.3542	13.62139	15.14325	14.52838	13.05388
		MIN	14.41382	13.00532	13.33584	14.98125	13.7833	12.91011
		STD	0.823848	0.493397	0.403824	0.229107	1.053705	0.20331
baboom	2	MAX	16.44527	16.20801	16.0649	16.05608	15.43013	16.00811
		MEAN	15.02325	14.98874	15.67748	15.31619	13.94943	15.28932
		MIN	13.98722	13.97204	15.20275	13.89877	12.56635	14.81462
		STD	1.188327	1.131664	0.437656	1.227915	1.434383	0.633079
lena	3	MAX	16.51593	16.54671	16.3681	16.06546	17.54182	16.42999
		MEAN	15.19492	16.37215	15.97784	15.04494	15.72654	16.09853
		MIN	13.87391	16.19759	15.58759	14.02443	13.91127	15.76706
		STD	1.868193	0.246871	0.551904	1.443222	2.567183	0.468761
baboom	3	MAX	18.33671	16.76244	18.3117	17.59036	17.06446	17.74262
		MEAN	16.54481	15.90312	18.12889	16.16509	16.85798	17.27462
		MIN	14.75291	15.04381	17.94607	14.73982	16.65149	16.80661
		STD	2.534129	1.215255	0.25854	2.015635	0.292014	0.661858
lena	4	MAX	18.67105	18.37671	18.69288	17.2902	18.26107	16.9732
		MEAN	17.97924	17.51948	17.80871	17.23195	17.33194	16.61104
		MIN	17.28743	16.66224	16.92454	17.1737	16.4028	16.24888
		STD	0.978372	1.21232	1.250405	0.082373	1.313999	0.512172
baboom	4	MAX	19.92586	19.37722	18.48785	18.67478	18.38887	19.10617
		MEAN	19.66968	18.19909	18.38787	18.00146	17.52779	17.78107
		MIN	19.4135	17.02097	18.28789	17.32814	16.66672	16.45597
		STD	0.362292	1.666119	0.141396	0.952217	1.217743	1.873974
lena	5	MAX	20.56957	18.53068	18.38683	19.70039	18.00139	19.41631
		MEAN	19.29008	18.20999	18.13655	17.84101	16.58188	18.08313
		MIN	17.91949	17.97141	17.78891	15.08727	14.11446	16.78843
		STD	1.327388	0.288535	0.310622	2.433131	2.144918	1.314362
baboom	5	MAX	21.13577	20.67446	19.48186	20.69898	19.59449	19.46577
		MEAN	20.94332	20.14955	18.61544	19.21363	18.90098	18.65874
		MIN	20.75086	19.62464	17.74901	17.72829	18.20747	17.85171
		STD	0.272171	0.742336	1.225313	2.100595	0.980771	1.141308

The fitness function values of all algorithms are mentioned and analyzed over the thresholds' variance. Tables 2, 3, 4 and 5 show the average fitness function values for the gray, red, green, blue images, where the threshold value is 2 for all algorithms. The proposed algorithm got better results in comparison with all the comparative algorithms. Whereas in levels 3, 4, and 5, the percentage of the difference is considerable between the MPASSA and other algorithms, as shown in Figs. 6 and 7 for grayscale images. Figure 8 shows the fitness values of the comparative methods for the different colored images. These results show the superiority of the MPASSA at the large threshold values.

In terms of PSNR measure, the testing results of the threshold two as in Tables 6 and 7, the MPASSA got the best results in all images (gray and color) followed by MPA at some times which got the best results, followed by WOA, PSO, SSA, and AOA. The results of threshold 2 indicate that the MPASSA obtained the best values in all images. The MPASSA outperformed all other algorithms in thresholds 3, 4, and 5. It achieved the best results in all images, followed by the WOA, MPA, PSO, and SSA. From the results, we can derive that MPASSA is the best algorithm according to the PSNR measure, followed by MPA and WOA, and these algorithms outperformed the others when the threshold is greater than 4.

Table 7 Comparison of *PSNR* for color images

Image	K	Measure	MPASSA	WOA	SSA	AOA	PSO	MPA
lena	2	MAX	14.4479	13.35268	14.27611	13.37208	14.28466	13.83051
		MEAN	12.92788	12.90309	13.71295	11.9712	13.54215	13.31247
		MIN	12.12286	12.5072	13.01171	10.79971	12.53165	12.51123
		STD	1.317146	0.425288	0.643407	1.301433	0.906716	0.703758
baboom	2	MAX	15.60419	14.7171	14.40342	15.35559	13.53726	14.08897
		MEAN	15.06041	13.39519	14.03065	14.73438	13.41879	12.84906
		MIN	14.51663	12.07329	13.65788	14.11317	13.30032	11.60915
		STD	0.769022	1.869459	0.52718	0.878523	0.167541	1.753501
peppers	2	MAX	15.35446	14.57942	15.94396	13.82622	13.59126	14.54233
		MEAN	14.62169	14.05543	15.58247	13.30836	13.42599	13.76463
		MIN	13.88891	13.53145	15.22099	12.7905	13.26072	12.98693
		STD	1.036305	0.741031	0.511219	0.732361	0.233731	1.099833
lena	3	MAX	17.80576	17.21374	16.57766	13.90539	15.94111	16.08695
		MEAN	14.93888	15.82893	14.90458	13.65691	14.88921	15.5409
		MIN	11.9958	13.70239	14.02877	13.47597	13.83199	14.45068
		STD	2.90573	1.869495	1.449466	0.222534	1.054574	0.944159
baboom	3	MAX	15.28153	15.28522	14.97328	13.48915	15.19341	13.74793
		MEAN	14.49329	14.39776	13.48258	13.0583	14.64112	13.20946
		MIN	1.114738	1.255065	2.108171	0.609316	0.781058	0.761501
		STD	0.710836	0.751651	1.336148	0.876276	0.5878	0.445365
peppers	3	MAX	17.56459	15.20602	16.30212	16.97933	15.91672	16.9768
		MEAN	16.37788	14.72653	16.18081	16.07364	14.93685	16.45331
		MIN	15.19117	14.24704	16.0595	15.16795	13.95698	15.92981
		STD	1.678263	0.678104	0.171558	1.280838	1.385743	0.740327
lena	4	MAX	19.12893	17.00327	16.22535	18.43391	17.54259	15.38833
		MEAN	17.67356	16.95789	15.94285	18.15588	16.27295	14.45334
		MIN	16.2182	16.91251	15.66035	17.87786	15.00331	13.51835
		STD	2.058199	0.064174	0.399516	0.393188	1.795539	1.322275
baboom	4	MAX	18.45459	16.73835	17.45467	14.9631	17.58325	17.02452
		MEAN	18.02775	16.32889	17.43965	14.31978	17.49489	16.10883
		MIN	17.60091	15.91944	17.42463	13.67647	17.40652	15.19314
		STD	0.603643	0.579058	0.021244	0.90978	0.124962	1.294976
peppers	4	MAX	18.74107	18.42268	18.38852	18.35256	17.75868	18.59458
		MEAN	18.46215	17.54987	18.26279	18.3015	17.69517	18.36283
		MIN	18.18324	16.67707	18.13706	18.25043	17.63166	18.13108
		STD	0.394448	1.234332	0.17781	0.072216	0.089815	0.327748
lena	5	MAX	22.23958	19.37222	20.29706	16.77697	18.51222	18.99759
		MEAN	19.74254	17.84263	18.35691	14.94191	17.17223	17.78203
		MIN	17.2455	16.31303	16.41677	13.10686	15.83224	16.56648
		STD	3.531342	2.163174	2.743784	2.595158	1.895031	1.719052
baboom	5	MAX	19.22274	18.05351	18.36224	18.02188	18.9775	17.56744
		MEAN	18.3831	17.81011	18.26134	17.33806	18.75711	17.437
		MIN	17.54345	17.56672	18.16044	16.65424	18.53673	17.30656
		STD	1.187435	0.344207	0.142696	0.967073	0.311673	0.18447
peppers	5	MAX	20.33411	18.09333	19.08781	19.6353	19.30731	19.14042
		MEAN	19.18374	17.17747	18.4008	18.58828	18.27825	17.93067
		MIN	17.72075	16.17611	17.0887	17.65838	17.45574	15.77134
		STD	1.334434	0.961468	1.13673	0.993653	0.942907	1.874554

In Tables 8 and 9, the comparative algorithms' results are presented in terms of SSIM for gray and color images, respectively. According to the results obtained by applying two thresholds, the proposed MPASSA got the best results in all images (i.e., gray and color), followed by SSA, which got the best results in 3 images out of 5, While the values of each

Table 8 Comparison of *SSIM* for gray images

Image	K	Measure	MPASSA	WOA	SSA	AOA	PSO	MPA
Lena	2	MAX	0.555608	0.470511	0.498464	0.528728	0.549838	0.445319
		MEAN	0.543559	0.462527	0.473189	0.516848	0.516584	0.438011
		MIN	0.531509	0.454543	0.447914	0.504969	0.483329	0.430703
		STD	0.01704	0.011291	0.035744	0.0168	0.047029	0.010335
baboom	2	MAX	0.608103	0.605281	0.596491	0.605382	0.57481	0.574172
		MEAN	0.57733	0.5558	0.586933	0.570285	0.507559	0.565484
		MIN	0.556117	0.521842	0.574179	0.50436	0.453318	0.560195
		STD	0.02728	0.043832	0.011494	0.057132	0.061782	0.007583
Lena	3	MAX	0.627357	0.595671	0.582347	0.530295	0.606453	0.561837
		MEAN	0.544744	0.586603	0.569464	0.507417	0.532994	0.543333
		MIN	0.462132	0.577535	0.556582	0.484538	0.459535	0.524829
		STD	0.116832	0.012824	0.018218	0.032355	0.103886	0.026169
baboom	3	MAX	0.705181	0.603365	0.676572	0.659079	0.662894	0.677553
		MEAN	0.619184	0.590509	0.668118	0.603442	0.641803	0.652842
		MIN	0.533187	0.577653	0.659664	0.547805	0.620712	0.628131
		STD	0.121618	0.018181	0.011956	0.078683	0.029827	0.034947
lena	4	MAX	0.665297	0.647772	0.617309	0.596659	0.634086	0.639709
		MEAN	0.657453	0.609306	0.578653	0.595772	0.624433	0.616709
		MIN	0.649609	0.57084	0.539998	0.594886	0.61478	0.593709
		STD	0.011093	0.0544	0.054668	0.001254	0.013651	0.032527
baboom	4	MAX	0.748646	0.694663	0.712826	0.69729	0.665902	0.700478
		MEAN	0.730368	0.664249	0.69375	0.670581	0.635046	0.66093
		MIN	0.712091	0.633836	0.674675	0.643872	0.60419	0.621382
		STD	0.025848	0.043011	0.026977	0.037772	0.043637	0.055929
lena	5	MAX	0.760291	0.653169	0.664039	0.609488	0.721824	0.686312
		MEAN	0.64427	0.639269	0.645897	0.562312	0.665422	0.632233
		MIN	0.519309	0.62615	0.633684	0.473536	0.610907	0.573331
		STD	0.12074	0.013527	0.016023	0.076933	0.055482	0.056645
baboom	5	MAX	0.798266	0.760516	0.724349	0.748132	0.747303	0.734778
		MEAN	0.790129	0.748593	0.701713	0.703145	0.722937	0.714183
		MIN	0.781992	0.736669	0.679076	0.658157	0.698572	0.693588
		STD	0.011507	0.016863	0.032013	0.063622	0.034458	0.029126

AOA, MPA, WOA and PSO algorithms are comparative according to the tested images. At the threshold level 3, the proposed MPASSA method obtained the best results in all images, followed by MPA, WOA, SSA, and AOA. Also, at the threshold levels 4 and 5, the proposed method obtained the best results in all images. Simultaneously, each WOA, SSA, AOA, PSO, and MPA are similar according to the SSIM values. From these results, we can conclude that MPASSA is the best algorithm among all comparative algorithms.

The CPU time is considered for all algorithms and listed in Tables 10 and 11 to show all algorithms' performance in terms of exhaustion time. To compare the time complexity of the proposed MPASSA method against other algorithms, Table 10 presents the CPU time for the gray level, and Table 11 presents the CPU time for the color images. These tables illustrated that the proposed method consuming average time between the other algorithms. It was ranked fourth, better than MPA and PSO algorithms, and worse than SSA, AOA, and WOA. While SSA was the lowers consuming of time execution followed by AOA and then WOA at all levels with all images (gray and color).

The Friedman ranking test results using PSNR and SSIM for all the tested methods using different tested images (gray and color) are given in Tables 12, 13, 14, 15, 16, 17, 18, 19, 20

Table 9 Comparison of *SSIM* for color images

Image	K	Measure	MPASSA	WOA	SSA	AOA	PSO	MPA
lena	2	MAX	0.795223	0.699592	0.74397	0.678005	0.706726	0.71598
		MEAN	0.646372	0.645989	0.716146	0.617766	0.683964	0.643045
		MIN	0.564897	0.596316	0.687505	0.517078	0.652683	0.583277
		STD	0.129102	0.05175	0.028241	0.087758	0.02801	0.067324
baboom	2	MAX	0.701557	0.632243	0.597474	0.679492	0.570098	0.58416
		MEAN	0.668994	0.535539	0.590432	0.645066	0.546319	0.49542
		MIN	0.636431	0.438836	0.583391	0.61064	0.52254	0.40668
		STD	0.046051	0.136759	0.009958	0.048686	0.033629	0.125498
peppers	2	MAX	0.746274	0.66342	0.708764	0.677896	0.648166	0.663263
		MEAN	0.736031	0.631924	0.69445	0.621832	0.639807	0.653526
		MIN	0.725788	0.600429	0.680135	0.565767	0.631448	0.643788
		STD	0.014485	0.044541	0.020244	0.079287	0.011821	0.013771
lena	3	MAX	0.864802	0.827958	0.862308	0.664223	0.811183	0.829273
		MEAN	0.745901	0.784284	0.772021	0.645614	0.731956	0.762389
		MIN	0.588434	0.702932	0.676012	0.611958	0.604451	0.682031
		STD	0.142163	0.070516	0.093279	0.029201	0.111502	0.07454
baboom	3	MAX	0.672814	0.654732	0.620069	0.544068	0.652885	0.549168
		MEAN	0.622775	0.602676	0.535066	0.527712	0.611122	0.524361
		MIN	0.070765	0.073618	0.120213	0.02313	0.059061	0.035084
		STD	0.030908	0.040504	0.085545	0.050867	0.031271	0.014398
peppers	3	MAX	0.784028	0.670445	0.764862	0.770122	0.748615	0.767455
		MEAN	0.780159	0.658666	0.74604	0.742439	0.707549	0.740247
		MIN	0.77629	0.646887	0.727218	0.714757	0.666482	0.713039
		STD	0.005472	0.016658	0.026618	0.039149	0.058077	0.038478
lena	4	MAX	0.907388	0.845001	0.774783	0.88183	0.826238	0.817124
		MEAN	0.862817	0.844678	0.770935	0.879423	0.756773	0.743345
		MIN	0.818246	0.844355	0.767086	0.877015	0.687308	0.669566
		STD	0.063033	0.000457	0.005442	0.003404	0.098238	0.104339
baboom	4	MAX	0.831887	0.72641	0.791766	0.632437	0.777011	0.753629
		MEAN	0.804432	0.706293	0.78796	0.585991	0.764687	0.713597
		MIN	0.776977	0.686177	0.784155	0.539545	0.752364	0.673566
		STD	0.038827	0.028449	0.005381	0.065685	0.017428	0.056613
peppers	4	MAX	0.845992	0.858013	0.835577	0.814155	0.851189	0.83603
		MEAN	0.839692	0.825313	0.820158	0.802811	0.825105	0.825229
		MIN	0.833393	0.792613	0.80474	0.791466	0.799021	0.814428
		STD	0.008909	0.046245	0.021805	0.016043	0.036889	0.015274
lena	5	MAX	0.944302	0.880396	0.919992	0.849059	0.811372	0.881303
		MEAN	0.879971	0.84098	0.865775	0.73073	0.765179	0.806263
		MIN	0.81564	0.801564	0.811559	0.6124	0.718985	0.731224
		STD	0.090978	0.055742	0.076674	0.167343	0.065327	0.106122
baboom	5	MAX	0.835687	0.782222	0.81866	0.787031	0.816234	0.763864
		MEAN	0.821109	0.778381	0.786501	0.763233	0.797111	0.759163
		MIN	0.806531	0.77454	0.754342	0.739434	0.777987	0.754462
		STD	0.020616	0.005432	0.04548	0.033656	0.027045	0.006648
peppers	5	MAX	0.879197	0.836983	0.866451	0.864195	0.861558	0.891493
		MEAN	0.852131	0.793081	0.861668	0.85134	0.849863	0.838455
		MIN	0.819286	0.750646	0.855161	0.841664	0.842271	0.774238
		STD	0.030371	0.043187	0.005839	0.011597	0.010277	0.059422

and 21. The Tables show the rank of each method, summation, mean ranking, and the final ranking.

For the gray images, in Table 12 of PSNR value, the MPASSA got the first ranking followed by WOA, it got the second-ranking, and SSA has the third ranking. Simultaneously,

Table 10 Average CPU time (seconds) for gray images

	K	WOA	SSA	AOA	MPA	PSO	MPASSA
Lena	2	0.251967	0.245896	0.239373	5.940693	1.146158	1.91396
	3	0.232458	0.228359	0.230218	7.630269	1.79152	2.651281
	4	0.241309	0.231332	0.238687	5.203141	2.080605	3.733159
	5	0.25015	0.245332	0.247421	11.48492	2.594214	4.058729
baboom	2	0.311161	0.245086	0.248864	2.90873	0.723672	2.736707
	3	0.291892	0.263487	0.249791	4.504445	1.12395	4.09117
	4	0.309512	0.303814	0.263115	10.71171	1.121382	4.379006
	5	0.31204	0.259287	0.266202	5.274792	1.188409	5.35945

Table 11 Average CPU time (seconds) for color images

	K	WOA	SSA	AOA	MPA	PSO	MPASSA
Lena	2	0.040097	0.03062	0.036374	76.88261	11.06229	6.020129
	3	0.044229	0.035656	0.043114	106.0657	15.30059	7.626052
	4	0.122027	0.064104	0.077583	175.7069	27.53357	16.09753
	5	0.113222	0.082588	0.091326	140.0356	26.58181	13.93868
baboom	2	2.307623	0.087376	0.079854	22.60579	3.105797	9.946913
	3	0.061295	0.047198	0.055619	101.6172	17.13897	7.643711
	4	0.061844	0.045043	0.054286	129.8905	23.27711	10.81031
	5	0.082559	0.077439	0.081497	28.96452	6.003071	20.01818
peppers	2	0.055243	0.042374	0.050997	79.58729	10.50436	6.968161
	3	0.043226	0.038953	0.046039	105.7169	17.60654	9.140672
	4	0.05905	0.041318	0.049201	131.4959	23.11287	10.81287
	5	0.058529	0.046055	0.058149	21.57042	32.8968	13.90128

the AOA and PSO ranked fourth, and MPA got the last ranking. These results proved the ability of the proposed MPASSA in solving the image segmentation problem. In Table 13 of the PSNR value, the MPASSA and SSA got the first ranking followed by AOA; it got the second-ranking, MPA has the third ranking. In contrast, WOA got the fourth ranking, and PSO has the fifth the last ranking in the table. This result illustrated that the proposed MPASSA has a promising outcome compared to other methods.

Table 12 The results of the Friedman ranking test for the comparative methods using the PSNR values for image lena

PSNR	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Lena	2	2	5	4	1	3	6
	3	5	1	3	6	4	2
	4	1	3	2	5	4	6
	5	1	2	3	5	6	4
	SUM	9	11	12	17	17	18
	Mean rank	2.25	2.75	3	4.25	4.25	4.5
	Final Rank	1	2	3	4	4	5

Table 13 The results of the Friedman ranking test for the comparative methods using the PSNR values for image baboom

PSNR	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Baboom	2	4	5	1	2	6	3
	3	4	6	1	5	3	2
	4	1	3	2	4	6	5
	5	1	2	6	3	4	5
	SUM	10	16	10	14	19	15
	Mean rank	2.5	4	2.5	3.5	4.75	3.75
	Final Rank	1	4	1	2	5	3

Table 14 The results of the Friedman ranking test for the comparative methods using the PSNR values for image lena

PSNR	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Lena	2	4	5	1	6	2	3
	3	3	1	4	6	5	2
	4	2	3	5	1	4	6
	5	1	3	2	6	5	4
	SUM	10	12	12	19	16	15
	Mean rank	2.5	3	3	4.75	4	3.75
	Final Rank	1	2	2	5	4	3

Table 15 The results of the Friedman ranking test for the comparative methods using the PSNR values for image baboom

PSNR	K	MPASSA	WOA	SSA	AOA	PSO	MPA
baboom	2	1	5	3	2	4	6
	3	2	3	4	6	1	5
	4	1	4	3	6	2	5
	5	2	4	3	6	1	5
	SUM	6	16	13	20	8	21
	Mean rank	1.5	4	3.25	5	2	5.25
	Final Rank	1	4	3	5	2	6

Table 16 The results of the Friedman ranking test for the comparative methods using the PSNR values for image peppers

PSNR	K	MPASSA	WOA	SSA	AOA	PSO	MPA
peppers	2	2	3	1	6	5	4
	3	2	6	3	4	5	1
	4	1	6	4	3	5	2
	5	1	6	3	2	4	5
	SUM	6	21	11	15	19	12
	Mean rank	1.5	5.25	2.75	3.75	4.75	3
	Final Rank	1	6	2	4	5	3

Table 17 The results of the Friedman ranking test for the comparative methods using the SSIM values for image lena

SSIM	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Lena	2	1	5	4	2	3	6
	3	3	1	2	6	5	4
	4	1	4	6	5	2	3
	5	3	4	2	6	1	5
	SUM	8	14	14	19	11	18
	Mean rank	2	3.5	3.5	4.75	2.75	4.5
	Final Rank	1	3	3	5	2	4

Table 18 The results of the Friedman ranking test for the comparative methods using the SSIM values for image baboom

SSIM	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Baboom	2	2	5	1	3	6	4
	3	4	6	1	5	3	2
	4	1	4	2	3	6	5
	5	1	2	6	5	3	4
	SUM	8	17	10	16	18	15
	Mean rank	2	4.25	2.5	4	4.5	3.75
	Final Rank	1	5	2	4	6	3

Table 19 The results of the Friedman ranking test for the comparative methods using the SSIM values for image lena

SSIM	K	MPASSA	WOA	SSA	AOA	PSO	MPA
Lena	2	3	4	1	6	2	5
	3	4	1	2	6	5	3
	4	2	3	4	1	5	6
	5	1	3	4	6	5	2
	SUM	10	11	11	19	17	16
	Mean rank	2.5	2.75	2.75	4.75	4.25	4
	Final Rank	1	2	2	5	4	3

Table 20 The results of the Friedman ranking test for the comparative methods using the SSIM values for image baboom

SSIM	K	MPASSA	WOA	SSA	AOA	PSO	MPA
baboom	2	1	5	3	2	4	6
	3	1	3	4	5	2	6
	4	1	5	2	6	3	4
	5	1	4	3	5	2	6
	SUM	4	17	12	18	11	22
	Mean rank	1	4.25	3	4.5	2.75	5.5
	Final Rank	1	4	3	5	2	6

Table 21 The results of the Friedman ranking test for the comparative methods using the SSIM values for image peppers

SSIM	K	MPASSA	WOA	SSA	AOA	PSO	MPA
peppers	2	1	5	2	6	4	3
	3	1	6	2	3	5	4
	4	1	2	5	6	4	3
	5	2	6	1	3	4	5
	SUM	5	19	10	18	17	15
	Mean rank	1.25	4.75	2.5	4.5	4.25	3.75
	Final Rank	1	6	2	5	4	3

Tables 14, 15 and 16 shows the ranking results based on using PSNR values for color images. In Table 14, the MPASSA got the first ranking, followed by each of WOA and SSA, which have got the second-ranking, MPA has the third-ranking. In contrast, PSO has the fourth-ranking, and AOA got the last ranking in the table. In Table 15, the MPASSA got the first ranking followed by PSO; it got the second-ranking, SSA has the third-ranking, while

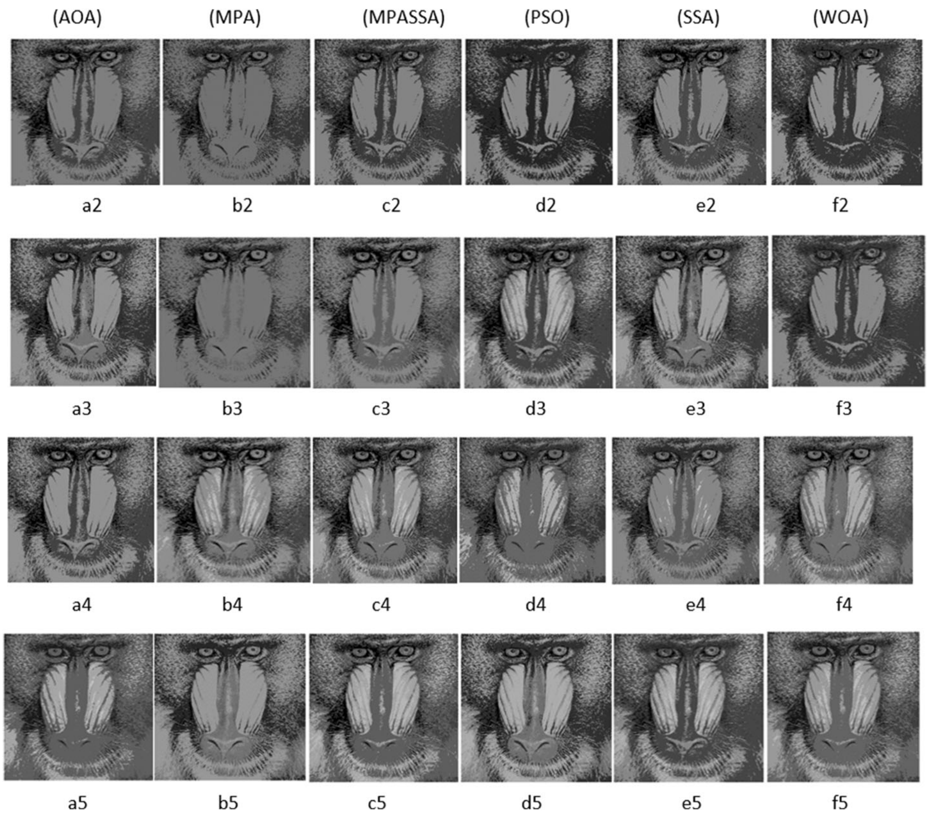

Fig. 9 Segmented *lena* gray images using Otsu's (a2–f2) represent threshold2, (a3–f3) represent threshold3, (a4–f4) represent threshold4, (a5–f5) represent threshold5

Fig. 10 Segmented *baboom* gray images using Otsu's (a2–f2) represent threshold2, (a3–f3) represent threshold3, (a4–f4) represent threshold4, (a5–f5) represent threshold5

WOA got the fourth-ranking, followed by AOA has the fifth ranking, and MPA at the last ranking in the table. In Table 16, the MPASSA got the first ranking, followed by SSA. It got the second-ranking, MPA has the third-ranking, while AOA got the fourth-ranking, followed by PSO have the fifth ranking and WOA at the last ranking in the table. The given results revealed that the modified version of MPA using SSA got better results than other comparative methods and avoided the basic version of MPA's weaknesses.

For the gray images in Table 17 of SSIM value, the MPASSA got the first ranking followed by PSO; it got the second-ranking, each of WOA and SSA has the third-ranking. In contrast, MPA has the fourth-ranking, and AOA got the last ranking. In Table 18 of SSIM value, the MPASSA got the first ranking followed by SSA; it got the second-ranking, MPA has the third-ranking. In contrast, AOA got the fourth-ranking, and WOA has the fifth ranking and PAO at the last ranking. This result illustrated that the proposed MPASSA has a better performance compared to other methods.

In Table 19, the MPASSA got the first ranking for the color images, followed by SSA, which got the second-ranking with WOA. At the same time, MPA has the third-ranking, and PSO got the fourth-ranking, and at the end ranking, AOA has the last ranking. In Table 20, the MPASSA got the first ranking followed by PSO; it got the second-ranking, SSA has the third-

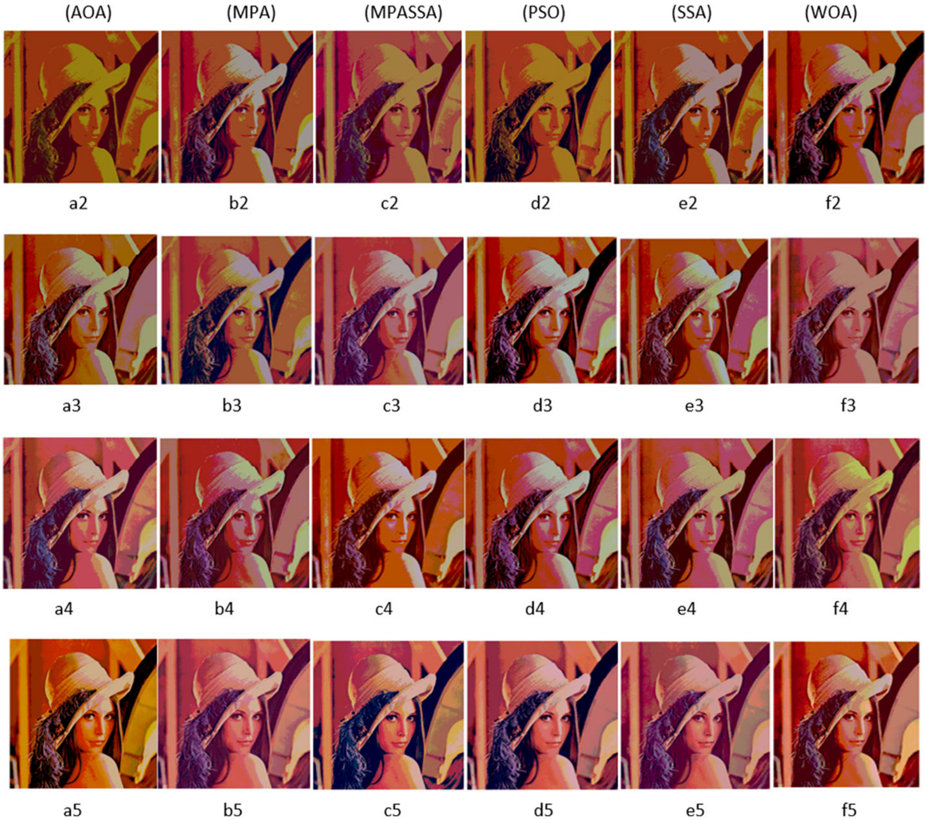

Fig. 11 Segmented *lena* color images using Otsu's (a2–f2) represent threshold2, (a3–f3) represent threshold3, (a4–f4) represent threshold4, (a5–f5) represent threshold5

ranking, while WOA got the fourth-ranking, followed by AOA has the fifth ranking and MPA at the last ranking. In Table 21, the MPASSA got the first ranking, followed by SSA; it got the second-ranking, MPA has the third-ranking, while PSO got the fourth-ranking, followed by AOA has the fifth ranking, and WOA at the last ranking. The given results showed that the proposed version of MPA using SSA got promising results compared to other comparative methods and avoided the basic version of MPA.

Figures 9, 10, 11, 12, and 13 show the segmented test images (gray and color) obtained by the comparative methods using various threshold values (k= 2, 3, 4, and 5). The proposed MPASSA got the best-segmented images compared to other comparative methods. Figure 14 shows the gray and color images histogram distribution estimated by the proposed hybrid algorithm MPASSA at k equal to 2, 3, 4, and 5.

The proposed MPASSA got the optimum results in most cases. This because each of the images is considered as a varied optimization problem. Moreover, because of the randomness of the swarm methods, the results can change in some cases. Also, depending on No-Free Lunch Theorem NFL that illustrates, it is hard to find one algorithm suitable for many optimization problems. As well as, each image has a varied gray or colored level histogram. Also, image segmentation becomes a complex operation due to

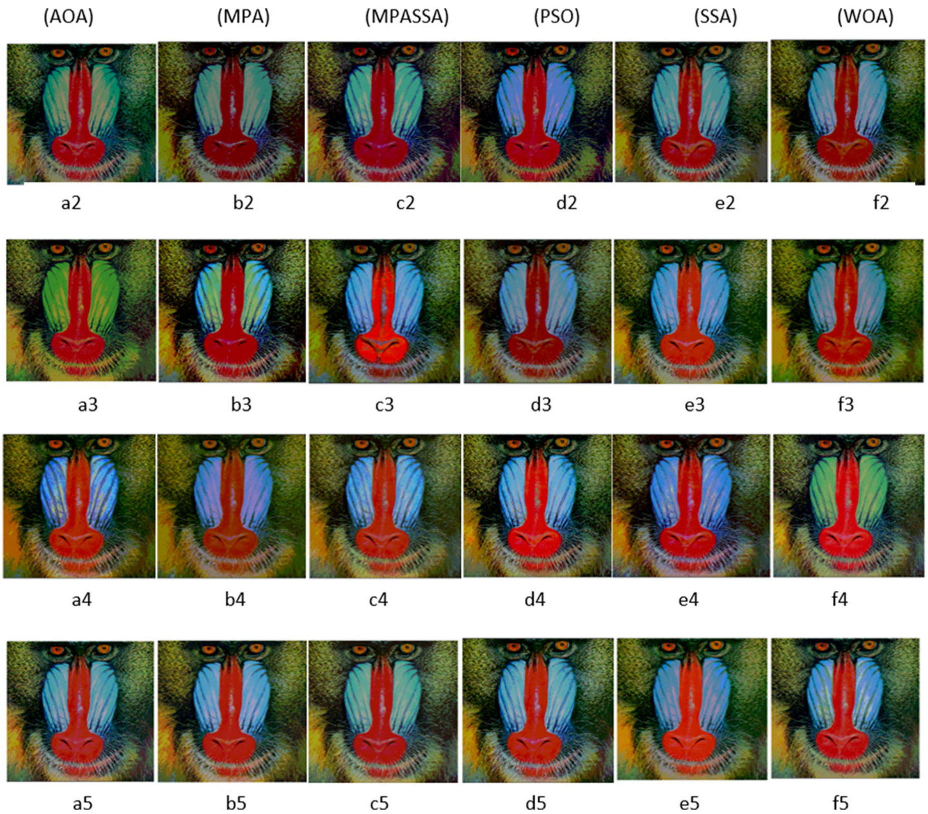

Fig. 12 Segmented *baboom* color images using Otsu's (a2–f2) represent threshold2, (a3–f3) represent threshold3, (a4–f4) represent threshold4, (a5–f5) represent threshold5

the histogram's multimodality. The segmented image is constructed by determining each class a grey or the colored level value specified from the class members' mean. So, for each image, the proposed MPASSA performed better in terms of the PSNR, SSIM, and fitness values at each level. The MPASSA may fail to segment the image in some cases and at different levels. For these reasons, we used statistical comparisons to determine the best algorithm for several images.

5 Conclusion and future work

Multilevel thresholding is one of the methods used in image segmentation. It is considered a preprocessing phase in many applications. In this research, the problem of defining the optimal/ best threshold for image segmentation in the case of multilevel thresholding. This problem has been considered as an 'optimization problem'. Otsu's function has been applied as a fitness function. So, a novel hybrid algorithm in image segmentation of multilevel threshold was proposed. It draws from the properties of both Marine Predators and Salp Swarm algorithms. The proposed algorithm has been used to solve this problem. Each one aims to define the optimal threshold value, which increases Otsu's function. The proposed

Fig. 13 Segmented *peppers* color images using Otsu's (a2–f2) represent threshold2, (a3–f3) represent threshold3, (a4–f4) represent threshold4, (a5–f5) represent threshold5

novel algorithm results have been compared with MPA, SSA, WOA, AOA, and PSA algorithms. The performance of algorithms has been evaluated based on the following measures: fitness values, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), CPU-time, and Friedman ranking test. We used five benchmark images for gray and color images in all experiments.

The study concluded that the proposed hybrid MPASSA method's performance is better than that of the MPA and SSA algorithms and other algorithms. The leading cause for this supremacy is that hybrid technology combines the characteristics of different methods. Therefore, the proposed hybrid algorithm averts getting stuck on a local optimum. All of this indicates that hybrid techniques have a high ability to find the optimal solution for the image segmentation problem.

The proposed method can solve various problems and applications as of image processing, such as visualization, computer vision, computer-aided diagnostics, image classification, etc. In all of these applications, the finesses of images are a critical problem in image segmentation. Low resolutions, high noises, or lower contrast, can cause this problem. So, the preprocessing of the images is an essential point to improve the quality of that status and multi-objective cases to gain optimal segmentation results. Other recent optimizers, such as Arithmetic Optimization Algorithm (AOA), can solve the given problem with new modifications in the future.

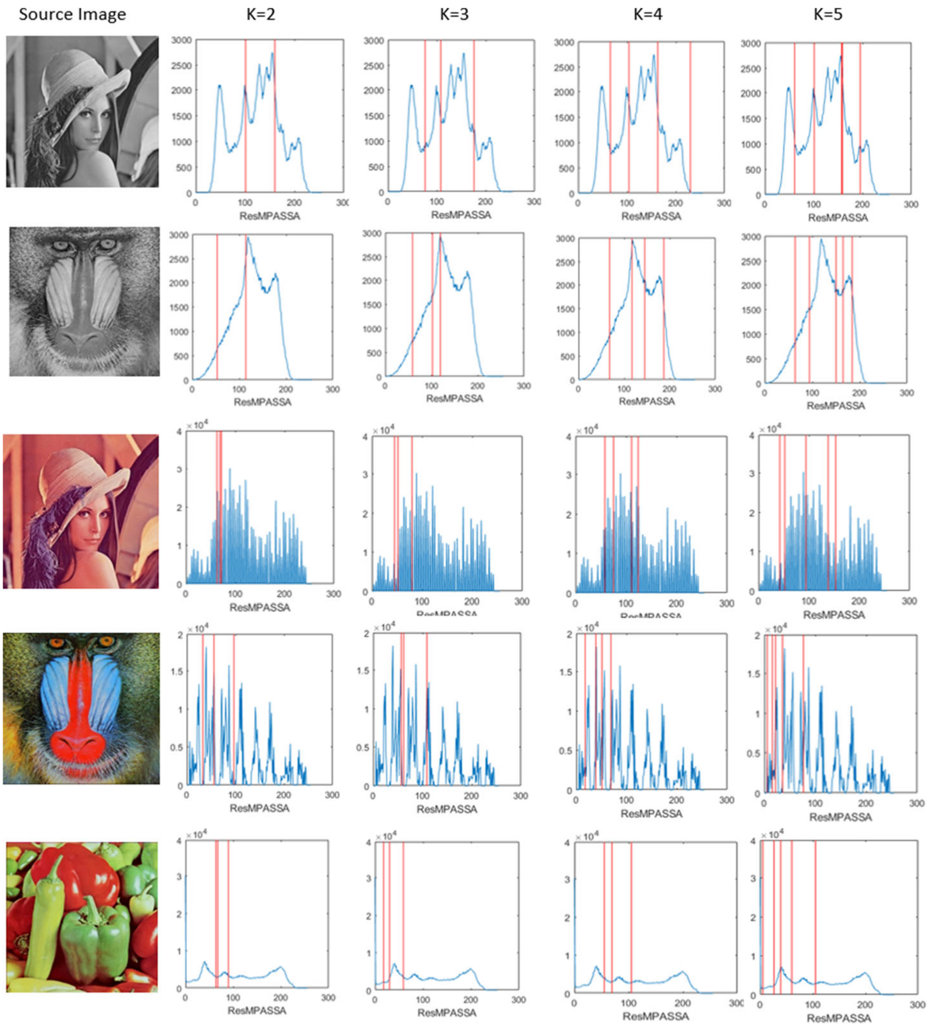

Fig. 14 Gy and Color images histogram at K=2, 3, 4, 5 obtained by MPASSA

Authors contribution Laith Abualigah: Conceptualization, methodology, writing, review, editing, and supervision. Nada Khalil Al-Okbi: Experimental, draft writing and validation. Mohamed Abd Elaziz: Experimental, Writing, review, editing, and supervision. Essam H. Houssein: Writing, review, and editing.**Data availability** - All the used data in this research will be available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no conflict of interest.

Human and animal rights This article does not contain any studies with human participants performed by any of the authors. No animal studies were carried out by the authors for this article.

References

1. Abd E, Aziz M, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256
2. Abd Elaziz M et al (2019) Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer. Expert Syst Appl 125:112–129
3. Abd Elaziz M et al (2020) An improved Marine Predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation. IEEE Access 8:125306–125330
4. Abd Elaziz M, Ewees AA, Oliva D (2020) Hyper-heuristic method for multilevel thresholding image segmentation. Expert Syst Appl 146:113201
5. Abd Elaziz M, Abualigah L, Attiya I (2021) Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments. Future Gener Comput Syst 124:142–154.
6. Abualigah L et al (2020) Salp swarm algorithm: a comprehensive survey. Neural Comput Appl 32(15): 11195–11215
7. Abualigah L et al (2020) Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis. Algorithms 13(12):345
8. Abualigah L et al (2021) Aquila Optimizer: A novel meta-heuristic optimization Algorithm. Comput Ind Eng 157:107250
9. Abualigah L (2021) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Comput Appl 33(7):2949–2972
10. Abualigah L et al (2021) A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of COVID-19 CT images. Processes 9(7):1155
11. Abualigah L et al (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376: 113609
12. Abualigah L, Alkhrabsheh M (2021) Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing. J Supercomput 78(1):740–765
13. Abualigah L, Diabat A (2020) A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl 32(19):15533–15556
14. Abualigah L, Diabat A (2021) Advances in sine cosine algorithm: a comprehensive survey. Artif Intell Rev 54(4):2567–2608
15. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795
16. Abualigah LM, Khader AT, Hanandeh ES (2018) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J Comput Sci 25:456–466
17. Abuowaida SFA, Chan HY, Alshdaifat NFF, Abualigah L (2021) A novel instance segmentation algorithm based on improved deep learning algorithm for multi-object images. Jordanian J Comput Inf Technol (JJCIT) 7(01). https://doi.org/10.5455/jjcit.71-1603701313
18. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13(6):3066–3091
19. Al-Qaness MA et al (2020) Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea. Int J Environ Res Public Health 17(10):3520
20. Altabeeb AM et al (2021) Solving capacitated vehicle routing problem using cooperative firefly algorithm. Appl Soft Comput 108:107403
21. An F-P, Liu J-e (2021) Medical image segmentation algorithm based on multilayer boundary perception-self attention deep learning model. Multimed Tools Appl 80(10):15017–15039
22. Bhandari AK et al (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(7):3538–3560
23. Dorgham O et al (2021) Monarch butterfly optimization algorithm for computed tomography image segmentation. Multimed Tools Appl: 1–34
24. Eid A, Kamel S, Abualigah L (2021) Marine predators algorithm for optimal allocation of active and reactive power resources in distribution networks. Neural Comput Appl: 1–29
25. Faramarzi A et al (2020) Marine Predators Algorithm: A nature-inspired metaheuristic. Expert Syst Appl 152:113377
26. Hassan MH et al (2021) Development and application of slime mould algorithm for optimal economic emission dispatch. Expert Syst Appl 182:115205
27. Hussien AG, Hassanien AE, Houssein EH (2017) Swarming behaviour of salps algorithm for predicting chemical compound activities. in eighth international conference on intelligent computing and information systems (ICICIS). IEEE
28. Ibrahim RA et al (2019) Improved salp swarm algorithm based on particle swarm optimization for feature selection. J Ambient Intell Humaniz Comput 10(8):3155–3169

29. Agushaka JO, Ezugwu AE, Abualigah L (2022) Dwarf Mongoose Optimization Algorithm. Comput Methods Appl Mech Eng 391:114570
30. Kapur JN, Sahoo PK, Wong AK (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Computer vision, graphics, and image processing, 29, p 273–2853
31. Kaur D, Kaur Y (2014) Various image segmentation techniques: a review. Int J Comput Sci Mob Comput 3(5):809–814
32. Khan Z, Yang J (2019) Image segmentation via multi dimensional color transform and consensus based region merging. Multimed Tools Appl 78(22):31347–31364
33. Martin D et al (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. IEEE
34. Mirjalili S et al (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
35. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
36. Mlakar U, Potočnik B, Brest J (2016) A hybrid differential evolution for optimal multilevel image thresholding. Expert Syst Appl 65:221–232
37. Mourchid Y, Hassouni ME, Cherifi H (2019) A general framework for complex network-based image segmentation. Multimed Tools Appl 78(14):20191–20216
38. Oliva D et al (2019) Multilevel thresholding by fuzzy type II sets using evolutionary algorithms. Swarm Evol Comput 51:100591
39. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
40. Ouadfel S, Taleb-Ahmed A (2016) Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Syst Appl 55:566–584
41. Phonsa G, Manu K (2019) A survey: image segmentation techniques. In: Harmony Search and Nature Inspired Optimization Algorithms. Springer, Berlin, pp 1123–1140
42. Premkumar M et al (2021) A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations. IEEE Access
43. Şahin CB, Abualigah L (2021) A novel deep learning-based feature selection model for improving the static analysis of vulnerability detection. Neural Comput Appl 33(20):14049–14067
44. Satapathy SC et al (2018) Multi-level image thresholding using Otsu and chaotic bat algorithm. Neural Comput Appl 29(12):1285–1307
45. Shubham S, Bhandari AK (2019) A generalized Masi entropy based efficient multilevel thresholding method for color image segmentation. Multimed Tools Appl 78(12):17197–17238
46. Singh KK, Singh A (2010) A study of image segmentation algorithms for different types of images. Int J Comput Sci Issues (IJCSI) 7(5):414
47. Soliman MA, Hasanien HM, Alkuhayli A (2020) Marine predators algorithm for parameters identification of triple-diode photovoltaic models. IEEE Access 8:155832–155842
48. Sumari P, Syed SJ, Abualigah L (2021) A novel deep learning pipeline architecture based on CNN to detect Covid-19 in chest X-ray images. Turk J Comput Math Educ (TURCOMAT) 12(6):2001–2011
49. Wang Q, Gao J, Yuan Y (2017) A joint convolutional neural networks and context transfer for street scenes labeling. IEEE Trans Intell Transp Syst 19(5):1457–1470
50. Wang Q, Gao J, Li X (2019) Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. IEEE Trans Image Process 28(9):4376–4386
51. Yin P-Y (2007) Multilevel minimum cross entropy threshold selection based on particle swarm optimization. Appl Math Comput 184(2):503–513
52. Yousri D et al (2020) A robust strategy based on marine predators algorithm for large scale photovoltaic array reconfiguration to mitigate the partial shading effect on the performance of PV system. IEEE Access 8: 112407–112426
53. Mahajan S, Abualigah L, Pandit AK, Altalhi M (2022) Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. Soft Comput 1–19.

Affiliations

Laith Abualigah[1,2] • Nada Khalil Al-Okbi[3] • Mohamed Abd Elaziz[4,5,6,7] • Essam H. Houssein[8]

Nada Khalil Al-Okbi
nada.prof1@gmail.com

Mohamed Abd Elaziz
abd_el_aziz_m@yahoo.com

Essam H. Houssein
essam.halim@mu.edu.eg

1 Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

2 School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia

3 Department of Computer Science, College of Science for Women, University of Baghdad, Baghdad, Iraq

4 Faculty of Computer Science & Engineering, Galala University, Suze 435611, Egypt

5 Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates

6 Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

7 School of Computer Science and Robotics, Tomsk Polytechnic University, Tomsk 634050, Russia

8 Faculty of Computers and Information, Minia University, 61519 Minia, Egypt