



Multilevel thresholding segmentation of color plant disease images using metaheuristic optimization algorithms

Rustu Akay¹ · Radhwan A. A. Saleh² · Shawqi M. O. Farea³ · Muzaffer Kanaan¹

Received: 27 March 2020 / Accepted: 17 August 2021 / Published online: 1 September 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

In this study, we apply multilevel thresholding segmentation to color images of plant disease. Given that thresholding segmentation is just an optimization problem, we use Otsu's function as the objective function. To solve this optimization problem, we implement five metaheuristic algorithms, namely artificial bee colony (ABC), cuckoo search (CS), teaching–learning-based optimization (TLBO), teaching–learning-based artificial bee colony (TLABC) and a modified version of TLABC proposed in this work, known as MTLABC. This version is a hybridization between TLABC and Levy flight where the search equations of TLABC are changed according to Levy flight equations; this modification, based on the experimental results, yields a significant improvement in TLABC. Various numbers of thresholding levels are tried to compare the performance of the optimization algorithms at multiple dimensions. The performance is measured according to five measures: the objective function, CPU time, peak noise-to-signal ratio, structural similarity index and color feature similarity. These measures indicate that our proposed algorithm, with the best values of the measures in most images and levels, ranks first. Also, Friedman and Wilcoxon signed-rank tests are used to analyze the results statistically. These two tests prove that our proposed algorithm is significantly different from the other four algorithms.

Keywords Multilevel thresholding segmentation · Metaheuristic optimization · Teaching–learning-based artificial bee colony · Plant disease

1 Introduction

Agricultural products play a key role in many economies worldwide. However, plant diseases are common and can occur naturally. Accordingly, these prevalent diseases are likely to adversely affect both the quantity and quality of these agricultural products. Then, the detection of plant diseases is crucial for that reason [25].

The main causes of plant disease are viruses, bacteria or even fungi. These various causes result in different visual symptoms, mainly on the plants. Fortunately, these

symptoms can be identified by the naked eye. Therefore, the conventional method for detecting plant diseases is the naked eye. Nonetheless, this method is not only time-consuming but also costly since a well-trained team of experts is required to conduct this type of examination [25].

An alternative automatic method based on image processing is proposed in the literature to overcome the traditional ways disadvantages. In essence, a diseased leaves image consists of a background, normal part and diseased part with visual symptoms, like a change in color [30]. So image segmentation is a key step in the process of plant disease detection such that the diseased part of the plant leaf can be extracted and examined to detect the disease and its degree if required. Indeed, segmentation is a crucial, inevitable step in image processing. The following image processing steps, such as feature extraction and pattern recognition, are highly dependent on this step [8].

In general, image segmentation divides the image into two or more parts according to the intensity of pixels,

✉ Radhwan A. A. Saleh
206113002@kocaeli.edu.tr

¹ Department of Mechatronics Engineering, Erciyes University, Kayseri, Turkey

² Mechatronics Engineering, Kocaeli University, Kocaeli, Turkey

³ Graduate School of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

where the pixels with similar intensity levels or similar properties are part of the same region. Although a wide range of image segmentation techniques exists in the literature, these techniques are generally divided into five main categories: clustering-based segmentation, matching-based segmentation, region-based segmentation, edge-based segmentation and thresholding-based segmentation [8]. Thresholding is considered as one of the most common segmentation techniques due to its robustness, simplicity and efficiency [20]. Thresholding-based segmentation methods divide into two broad types: bi-level and multi-level thresholding [20]. In bi-level thresholding, we try to find one threshold value that separates the image into a background and object. However, in multilevel thresholding, we try to find two or more threshold values to separate the image into distinct meaningful regions or objects. Consequently, multilevel thresholding demands more computation than bi-level thresholding and the computation complexity increases exponentially with the number of thresholds [20]. In the plant disease detection, we need to use multilevel thresholding rather than binary thresholding as an image of a diseased leaf contains at least three distinct objects: the background, normal part and diseased part.

Optimization is an important branch of mathematics that aims at finding the optimal solution of a given problem and under given constraints [10]. Thresholding is an optimization problem where we try to find the best threshold values minimizing some objective function. In terms of the optimization objective function, thresholding can be separated into two types: entropy-based thresholding and between-class variance-based thresholding. Kapur entropy and Otsu (between-class variance-based approach) are the most widely used objective functions [21]. In addition to these two common objective functions, Renyi entropy and Tsallis entropy are another candidate objective functions for the thresholding segmentation [19].

Deterministic optimization methods can be used for multilevel thresholding problems, but one of their major drawbacks is the computation time, particularly when the number of threshold values is high [20]. For instance, dynamic programming, which is one of the famous deterministic optimization algorithms, was used in [17] to solve the thresholding segmentation problem with Otsu's objective function. However, this algorithm's main limitation, like other deterministic optimization algorithms, is the computation time, which increases exponentially with the number of threshold values [19]. In [19], a modified version of the Otsu's between-class variance was used as the objective function for the deterministic optimization algorithm used in [17]. This modified Otsu's objective function and the original one share the same solution.

Metaheuristic algorithms are powerful optimization algorithms that are often used as alternative, efficient methods to the traditional optimization techniques to overcome their limitations [20]. Indeed, metaheuristic algorithms achieve better results than the traditional optimization algorithms [6]. In the literature, there are innumerable metaheuristic algorithms and new ones are constantly being developed [15]. One of them is the best depends on the application itself; that is, one algorithm might show good results in one specific application but show bad results in other applications [20]. Each algorithm has its limitations; some have good exploitation ability while the others have good exploration ability [6]. Hybrid optimization algorithms are proposed in the literature to overcome the drawbacks of the individual algorithms and mix their strengths.

In [21], nine of metaheuristic algorithms were listed as multilevel segmentation algorithms. These nine bio-inspired algorithms were ABC, CS, bat algorithm, social spider optimization, firefly algorithm, gray wolf algorithm, moth flame optimization, whale optimization algorithm and particle swarm optimization. A comparison between these algorithms was drawn based on four factors: the value of the objective function (both Otsu and Kapur were used as the objective function), peak signal-to-noise ratio (PSNR) [12], structural similarity index (SSIM) [26] and lastly CPU time.

In [1], five of the metaheuristic methods were used to solve the multilevel segmentation problem. These five algorithms were the particle swarm optimization, culture algorithm, genetic algorithm, artificial tree algorithm and CS. They were compared against each other in terms of the computation time, value of the objective function which was the Levine and Nazif intra-class uniformity and PSNR.

In [6], several stochastic algorithms, including the ABC, firefly algorithm, differential evolution, social spider optimization, particle swarm optimization, whale optimization algorithm, moth flame optimization, gray wolf optimization, multiverse optimization, selfish herd optimization and harmony search, were used and compared in solving the thresholding problem. Time complexity, fitness values, PSNR and SSIM were the performance factors used to assess the different algorithms. In addition, Wilcoxon's ranked-sum statistical test was carried out to analyze the results statistically.

In [4], a modified version of the particle swarm optimization algorithm was used as the optimization algorithm for multilevel thresholding with the recursive minimum cross-entropy as the objective function. This proposed algorithm was compared with the original CS, particle swarm algorithm, genetic algorithm, firefly algorithm and modified ABC. The comparison was drawn in terms of the following performance factors: values of the objective

function, computation time, misclassification error, complex wavelet structural similarity index measurement, feature similarity index (FSIM) and PSNR.

In [7], a hybrid algorithm of ABC and the sine–cosine algorithm was implemented for the multilevel segmentation, for which the Otsu’s function was used as the objective function. The performance of this hybrid algorithm was compared with the original ABC algorithm and sine–cosine algorithm according to the fitness values, CPU time, SSIM and PSNR. Additionally, [13] applied a novel metaheuristic algorithm, known as black widow optimization, to the multilevel thresholding segmentation. Both Otsu and Kapur were used as the objective function. The results were compared against those of a six other metaheuristic optimization algorithms according to SSIM, PSNR, FSIM as well as the fitness function.

All of the studies mentioned above tried to segment gray images using multilevel thresholding. Unlike these studies, a study reported in [11] used a metaheuristic algorithm called the efficient Krill Herd algorithm to segment color images using multilevel thresholding segmentation. In that study, the Tsallis entropy, Kapur and Otsu were used all as the objective function. In addition to that study, color satellite images were segmented in [24] using multilevel thresholding. Masi entropy was used as the objective function of the optimization problem and compared with the other famous entropy-based objective functions, including Kapur, Tsallis and Renyi entropy. Using Masi entropy [18] showed better results for color satellite images as compared to the other three entropy-based criteria. The performance was assessed based on PSNR, FSIM, SSIM, mean square error (MSE) and misclassification error.

The last study implemented satellite images, while the other studies mentioned above used some benchmark images. Below is some literature review about segmentation for plant disease detection. In [25], color images of five different leaf diseases were segmented using genetic algorithm, with the Euclidean distance between the pixels and their corresponding clusters as the fitness function. The five types of leaf diseases were frog eye leaf spot, early scorch, fungal disease Sunburn disease and bacterial leaf spot. In [2], K-means clustering method [9] was carried out to segmentation of color images of plant leaves. There were four clusters; that is, each image was segmented into four regions. Different values of the number of clusters were tested and the best results were achieved when the number of clusters was four. In [14], color images of diseased leaves were segmented by using K-means clustering and additional segmentation step was added in order to mask the mostly green-colored pixels. This additional segmentation was done using Otsu’s method.

In [27], a two-step segmentation of color images of diseased leaves was carried out using Otsu’s method as the

first step and Sobel operator as the second step. In fact, Otsu’s method was commonly used in the literature in order to segment images of diseased plants after some preprocessing had been implemented to the images. In [30], segmentation of color images of diseased leaves was implemented via K-means clustering and super-pixel clustering. The segmentation process went through two distinct steps. The first step was by applying super-pixel clustering; after that, K-means clustering of the super-pixel clustering-segmented images was conducted.

The contribution of this study can be summarized as follows. First, we introduce a novel modified version of TLABC. Then, we apply it the multilevel thresholding color image segmentation and compared our results with four other closely related metaheuristic algorithms: ABC, TLBO, TLABC and CS. The proposed version of TLABC introduces two modifications to the standard TLABC algorithm. The first one is the use of Levy flight equations in the scout phase while the second one is the addition of an indicator of the fertile area to the framework strategy. These two modifications, in fact, substantially improve the performance of the algorithm. The next section provides more details and explanations about these modifications.

The rest of this paper is as follows: Section 2 explains the methodology and algorithms used in this work, Sect. 3 sheds light on the different metrics used to assess the performance of the used algorithms, Sect. 4 displays the results and findings of this study with relevant discussion, and finally, Sect. 5 concludes this paper.

2 Methodology

Color image thresholding is considered a problem of optimization. In this research, metaheuristic algorithms have been used to solve this optimization problem. The objective function of this optimization problem is the Otsu’s function where the optimal thresholds separate the image pixels into different classes with minimal intra-class variance and maximal inter-class variance. That is, Otsu’s method is a maximization problem exhaustively searching for the best thresholds maximizing the inter-class variance and equivalently minimizing the intra-class variance. Metaheuristic algorithms, depend on an iterative process to find the optimal solution to the optimization problems. Each metaheuristic algorithm has its way of generating and replacing the solutions during the searching process, leading to different outputs. In this section, we will introduce ABC, CS, TLBO, TLABC and TLABC with Levy flight algorithms, a new modified version of TLABC proposed in this work, in detail.

2.1 Thresholding problem

The image thresholding problem searches for the optimal m levels such that the original image is segmented into $(m + 1)$ subimages. When m equals 1, it is called bi-level thresholding while it becomes multilevel thresholding when m is greater than one. Thus, the m – level thresholding is an optimization problem trying to find the optimal vector $[t_1, t_2, \dots, t_m]$ that minimizes some objective function (in our case, the Otsu function). That is, the original image I (with intensity $f(x, y)$ at the location (x, y)) is segmented into $(m + 1)$ subimages as follows:

$$\begin{aligned}
 I_o &= \{f(x, y) \in I | 0 \leq f(x, y) \leq t_1 - 1\} \\
 I_1 &= \{f(x, y) \in I | t_1 \leq f(x, y) \leq t_2 - 1\} \\
 &\vdots \\
 I_i &= \{f(x, y) \in I | t_i \leq f(x, y) \leq t_{i+1} - 1\} \\
 &\vdots \\
 I_m &= \{f(x, y) \in I | t_m \leq f(x, y) \leq 255\}
 \end{aligned}
 \tag{1}$$

While Eq. (1) normally applies to the gray-level images, it can apply to the red, green and blue channels of the RGB color images as well where each channel of the RGB image is treated as an individual image.

2.2 ABC algorithm

Among swarm intelligence optimization algorithms, the ABC algorithm is considered as one of the most effective algorithms due to its major advantages such as high exploration, simple implementation and less control parameters [16]. To improve its performance more and overcome its limitations like poor exploitation and slow convergence, many modified versions of this algorithm were proposed in the literature. A recently modified version along with other already existing modified versions of the ABC algorithm, were compared in [16].

The ABC algorithm divides the bee foraging task into three phases: employed and onlooker bees for the purpose of exploitation in addition to scout bees for the purpose of exploration. These three types of tasks lead the bees to converge toward the best food source by sharing information with each other. The bee colony is divided into two equal subsets of bees, the employed and onlookers. A scout bee carries out a random search. The employed bee becomes a scout as soon as its food source is exhausted. Each of these tasks is described in detail as follows.

Initialization Initial population is randomly generated using Eq. (2).

$$x_i^j = x_{min}^j + rand(0, 1)(x_{max}^j - x_{min}^j)
 \tag{2}$$

Employed Bee Task Equation (3) is used for performing the

update of the current solution where $k \in [2, SN]$ and $j \in [2, D]$ are randomly chosen indices, SN is the number of food sources, D is the dimension of the solution (in this paper, D is the number of levels) and k is not equal to i . Employed bees update the existing solution based on the fitness value of the new solution which is calculated by Eq. (3). Then they replace the existing food source with the new one of higher fitness value.

$$v_i^j = x_i^j + rand(0, 1)(x_i^j - x_k^j)
 \tag{3}$$

Onlooker Bee Task It starts when the employed bees phase finishes. In this task, the onlooker bees in the hive calculate the selection probability of food sources based on the employed bees that shared the new solutions. The probability may be computed using Eq. (4).

$$P_i = \frac{Fit(i)}{\sum_{i=1}^N Fit(i)}
 \tag{4}$$

Scout Bee Task A food source is assumed to be exhausted if that food source is not improved for a threshold number of iterations. Then employed bee becomes a scout, and a new food source will be created randomly inside the search space using Eq. (2).

2.3 CS algorithm

Due to its high efficiency in solving various types of optimization problems and real-world applications, CS has attracted great attention [28]. There are three idealized rules of simplicity in the definition of CS: (1) Each cuckoo lays only one egg in a randomly selected nest at a time. (2) Only bring the best nests to the next generations. (3) The number of available host nests is fixed, and the host bird can discover the egg laid by a cuckoo with a probability $pa \in [0, 1]$.

$$x'_{ij}(t + 1) = x_{ij}(t) + stepsize(t) \oplus Levy
 \tag{5}$$

2.4 TLBO algorithm

The algorithm TLBO mimics a teacher’s effect on students. There are two stages in the TLBO algorithm: teacher phase and student phase [22].

Teacher phase The best solution from the population is known as the teacher since the teacher is considered the most knowledgeable person in the class. The teacher provides learners with information to maximize the mean outcome of the lesson using Eq. (6). This phase is responsible for the exploration process.

$$x_{i,d}^{new} = x_{i,d}^{old} + rand_2(x_{teacher,d} - T_F * x_{mean,d}) \tag{6}$$

Learner phase Through communicating with each other, the learners increase their knowledge. If one of the students has more experience and knowledge than the others, then all other learners will learn new information. This phase is the responsibility of the exploitation process.

$$u_s = \begin{cases} x_s + rand \cdot (x_s - x_j), & \text{if } f(x_s) \leq f(x_j) \\ x_s + rand \cdot (x_j - x_s), & \text{if } f(x_j) > f(x_s) \end{cases} \tag{7}$$

2.5 TLABC algorithm

TLABC is hybridization between TLBO and ABC algorithms which combines the advantages of both (the exploration of ABC and the exploitation of TLBO). It effectively employs three hybrid search phases as follows [5].

Teacher phase Here, each employed bee uses a hybrid of TLBO and mutation operator of differential evolution to search a new food source, which can develop the variety of search tendencies extraordinarily and upgrade the searchability of TLABC.

Learning-Based Onlooker Bee Phase In this stage, an onlooker bee chooses a sustenance source to search out as indicated by the selection probability, which is determined utilizing Eq. (8). After that, the onlooker bee finds out new food sources using the TLBO’s learning strategy.

$$u_{i,d} = \begin{cases} x_{i,d}^{old} + rand_2(x_{teacher,d} - T_F \cdot x_{mean,d}), & \text{if } rand_1 < 0.5 \\ x_{r1,d} + F \cdot (x_{r2,d} - x_{r3,d}), & \text{otherwise} \end{cases} \tag{8}$$

Generalized Oppositional Scout Bee Phase In this stage, if a nourishment source cannot be improved further for a specific period, it is viewed as depleted and would be relinquished. At that point, an arbitrary candidate solution and the generalized oppositional solution of it are created. The best solution of them is utilized rather than the old depleted nourishment source.

$$x_{ij}^{GO} = k \cdot (a_j + b_j) - x_{ij} \tag{9}$$

2.6 The proposed modification of TLABC algorithm (MTLABC)

TLABC has a good balance between exploration and exploitation, as described above. Yet, the modification of TLABC proposed in this paper would significantly improve its efficiency. The TLABC algorithm was modified in previous work; the framework of TLABC was maintained while the equations of searching of onlooker and employed phases were modified [23]. Nonetheless, the currently

proposed modification of the TLABC algorithm is, in essence, a kind of low-level integrative hybridization between the TLABC and the Levy flight algorithm. In fact, the good exploration ability of the ABC algorithm attracts a considerable number of researchers to hybridize it with many other swarm algorithms, as mentioned previously. Some of them keep both the search equations and the ABC framework while the others keep just the ABC framework. In the TLABC hybridization, only the framework was maintained; therefore, in this paper, we have also modified the search equation used in the scout phase. This modification enhances the performance of exploitation searching of TLABC in the generalized oppositional scout bee phase; the solution created randomly using Eq. (2) was replaced by a solution created using the update equation of Levy flight using Eq. (6) around the fertile area of the search rather than the old depleted nourishment source. In fact, the scout phase of the original ABC and TLABC generates solutions randomly to improve the exploration more. However, the exploration of ABC is already high and we here make some trade-off between the exploration and exploitation by generating solutions near the best solution instead of randomly. By doing so, we improve the exploitation of the algorithm and achieve some type of balance between exploration and exploitation. In addition, we add an indicator of the fertile area which is the other modification we add to the TLABC. The fertile area could be discovered by using this indicator to each solution which will increase whenever the solution increases. Thus, the solution which its indicator is the highest is indeed in the fertile area. The pseudo-code for this change is shown in Algorithm (1). The parameter settings of the MTLABC algorithm are listed in Table 1.

Table 1 Parameters setting of MTLABC algorithm

Parameter	Value
The initialization within the search space was uniform random	
Number of iteration	100
Limit	100
Colony size NP	100
Number of food sources	$SN = NP/2$
The range search was inside the interval	$[0, 255]$
pa	0.5

Algorithm 1 The pseudo code of MTLABC Algorithm

```

1: Initialize the population by Eq. (2),
2: Calculate the fitness values of the population using Eq. (3),
3: Set  $trail(i) = 0$  and  $indicator(i) = 0$  for each individual,
4: repeat
5:   for  $i = 1 \dots NP$  do
6:     Move the employed bees by using Eq. (8),
7:     if the new food source is better then
8:       Replace solutions ( $x_{i,d} = u_{i,d}$ ),
9:        $indicator(i) = indicator(i) + 1$ ,
10:       $trail(i) = 0$ ,
11:     end if
12:   end for
13:   for  $i = 1 \dots NP$  do
14:     Calculate selection probability using Eq. (2),
15:     Move the onlooker bees by using in Eq. (7)
16:     if the new food source is better then
17:       Replace solutions ( $x_{i,d} = u_{i,d}$ ),
18:        $indicator(i) = indicator(i) + 1$ ,
19:        $trail(i) = 0$ ,
20:     end if
21:   end for
22:   if The trail is greater than the limit then
23:     Find the solution with the maximum indicator
24:     Update its value by applying the update Eq. (5)
25:     Create its generalized oppositional sol. using Eq. (9)
26:   end if
27:   Memorized the best solution
28: until (Cycle=MaxCycle or termination criteria are met)

```

Teaching based employed bee phase
 Learning based onlooker bee phase
 Generalized oppositional scout bee phase

3 Image quality assessment (IQA) metrics

Loss of information, together with a significant degradation in the quality of an image, occurs at different stages of its life cycle, including acquisition, transmission, restoration, processing or compression [12, 26]. Therefore, various methods, known as image quality assessment, have developed in the literature to assess the quality of a particular image after going through some type of distortion. In general, IQA methods divides into two main approaches: subjective and objective approaches. In the subjective methods, the image quality is evaluated by a human. However, this method is costly, time-consuming and subjective (can differ according to the person evaluating the image). On the other hand, the objective approaches assess the image according to some quantitative measures so the image quality assessment can be done automatically using some algorithms and metrics. Some of the most widely used objective IQA metrics are PSNR, SSIM, FSIM and color feature similarity (FSIMc) indices [29].

3.1 PSNR index

PSNR, which is a function of the MSE, is considered as the simplest IQA metric. It is very straightforward to compute, mathematically convenient for optimization, and has a

clear physical meaning. MSE measures the mean-squared error between the intensities of the reference image pixels (the image before segmentation in our case) and the distorted image pixels (the image after segmentation in our case). The PSNR index is calculated according to Eq. (10).

$$PSNR(I_R, I_S) = 10 \log_{10} \left(\frac{255^2}{MSE(I_R, I_S)} \right) \quad (10)$$

where I_R and I_S are the original image and the segmented image, respectively. The MSE is obtained using Eq. (11).

$$MSE(I_R, I_S) = \frac{1}{m * n} \sum_{i=1}^m \sum_{j=1}^n [I_R(i, j) - I_S(i, j)]^2 \quad (11)$$

where the size of the image is $m - by - n$, and the MSE is averaging the sum of the difference between the original and segmented images squared.

The PSNR index measures the quality of the segmented image as compared to the original one in decibels. As inferred from Eq. (10), when the MSE approaches zero, the PSNR goes to infinity. Consequently, high values of the PSNR mean a high quality of the segmented image and vice versa [12]. Although the MSE and PSNR are easy to compute, both of them fail to distinguish structural content of the image. In fact, two images with different levels of degradation can have the same MSE (PSNR) values.

3.2 SSIM index

As mentioned before, the MSE and PSNR fail to discriminate the structural content of an image. Nevertheless, the nature of images is highly structured where proximate pixels of an image are highly dependent and these dependencies are due to the structure of the different objects on the image. Thus, SSIM tries to measure the differences between the structural content of the original image and the distorted image (the segmented image in this paper). The SSIM index imitates how the human visual system (HVS) assesses the image quality depending upon the structures of the images. The SSIM index models the image distortion as three distinct factors: luminance distortion, correlation distortion and contrast distortion. Then, the SSIM index is defined as follows:

$$SSIM(I_R, I_S) = L(I_R, I_S) \cdot C(I_R, I_S) \cdot S(I_R, I_S) \tag{12}$$

$$L(I_R, I_S) = \frac{2 \cdot \mu_{I_R} \mu_{I_S} + c_1}{\mu_{I_R}^2 + \mu_{I_S}^2 + c_1} \tag{13}$$

$$C(I_R, I_S) = \frac{2 \cdot \sigma_{I_R} \sigma_{I_S} + c_2}{\sigma_{I_R}^2 + \sigma_{I_S}^2 + c_2} \tag{14}$$

$$S(I_R, I_S) = \frac{\sigma_{I_R I_S} + c_3}{\sigma_{I_R} \sigma_{I_S} + c_3} \tag{15}$$

where I_R and I_S are the original (reference) image and the segmented image, respectively. $L(I_R, I_S)$ is the luminance function which compares the similarity between the luminance means of the original and segmented images. This function is maximum and equal to 1 when $\mu_{I_R} = \mu_{I_S}$. $C(I_R, I_S)$ is the contrast function measuring the similarity between the original and segmented images. This function is maximum and equal to 1 when the standard deviations of the original and segmented images are equal $\sigma_{I_R} = \sigma_{I_S}$. $S(I_R, I_S)$ is the correlation or structural function which measures the correlation between the two images where $\sigma_{I_R I_S}$ is the covariance between the original and segmented images. The higher the SSIM index is, the higher the quality of the segmented image is. The three constants in the above equations are used to ensure stability and make sure the denominators are always nonzero.

3.3 FSIM index

FSIM is other IQA metric which evaluates the image quality relying upon its low-level features, just as HVS does. These low-level features are the phase congruency (PC) of the image and the gradient magnitude (GM). The primary feature of the FSIM index is the PC, which is a dimensionless indicator of the significance of the local structure. Although the PC is a reasonable model of how HVS identifies the image features, it is contrast-invariant

given that the HVS perception of the image quality does also depend on the information about the contrast in the image. So the GM is the secondary feature of the FSIM index which takes into account the local contrast to play a complementary role with the PC in characterizing the image local structure [29]. The process of computing the FSIM index consists of two steps:

Step 1: The map of the local similarity between the two images (the original and segmented images) is calculated. In fact, the PC and GM are two components determining the local similarity map. Therefore, the similarity between the original and segmented images in terms of their PC's is calculated as in Eq. (16).

$$S_{PC} = \frac{2 \cdot PC_1 \cdot PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1} \tag{16}$$

where PC_1 and PC_2 are the PC of the original and segmented images and T_1 is a positive constant to increase the stability and ensure the denominator does not equal zero (the value of T_1 relies upon the dynamic range of the PC). The values of S_{PC} are in the range (0, 1] such that small values near zero indicate that the PC of the two images are different and high values near one indicate that the PC of the two images are similar. Likewise, the similarity between the original and segmented images in terms of their GM's is calculated according to Eq. (17).

$$S_G = \frac{2 \cdot G_1 \cdot G_2 + T_2}{G_1^2 + G_2^2 + T_2} \tag{17}$$

where G_1 and G_2 are the GM of the original and segmented images and T_2 , like T_1 , is a positive constant calculated according to the dynamic range of the GM.

Step 2: The two components, the PC and GM, are combined to come up with a single value according to Eq. (18).

$$S_L = (S_{PC})^\alpha \cdot (S_G)^\beta \tag{18}$$

where S_L is the combined similarity between the original and segmented images. α and β are two parameters weighing the importance of the GM and PC. For simplicity, both α and β are normally put equal to 1. After obtaining S_L , the overall similarity between the original and segmented images is:

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x) \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \tag{19}$$

Ω is the whole image spatial domain. S_L at any location x is weighted by $PC_m(x)$ where $PC_m(x) = \max(PC_1, PC_2)$. So the PC acts as a weight because different locations have different importance in determining the similarity between two images according to the HVS. Additionally, the PC structures determine the significance of each location. Like

Fig. 1 Eight images and their histograms

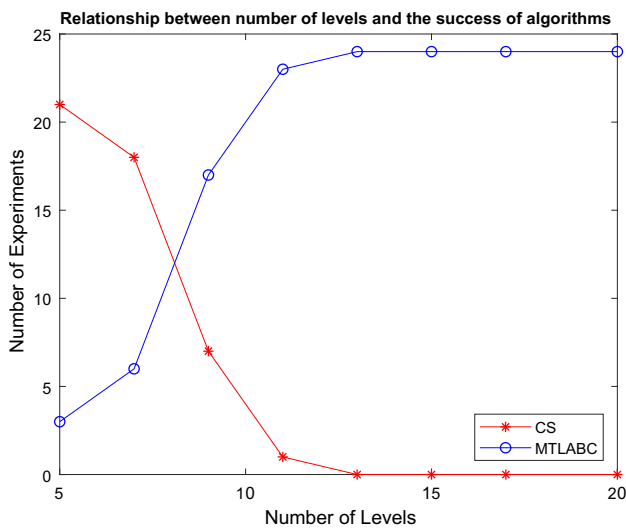
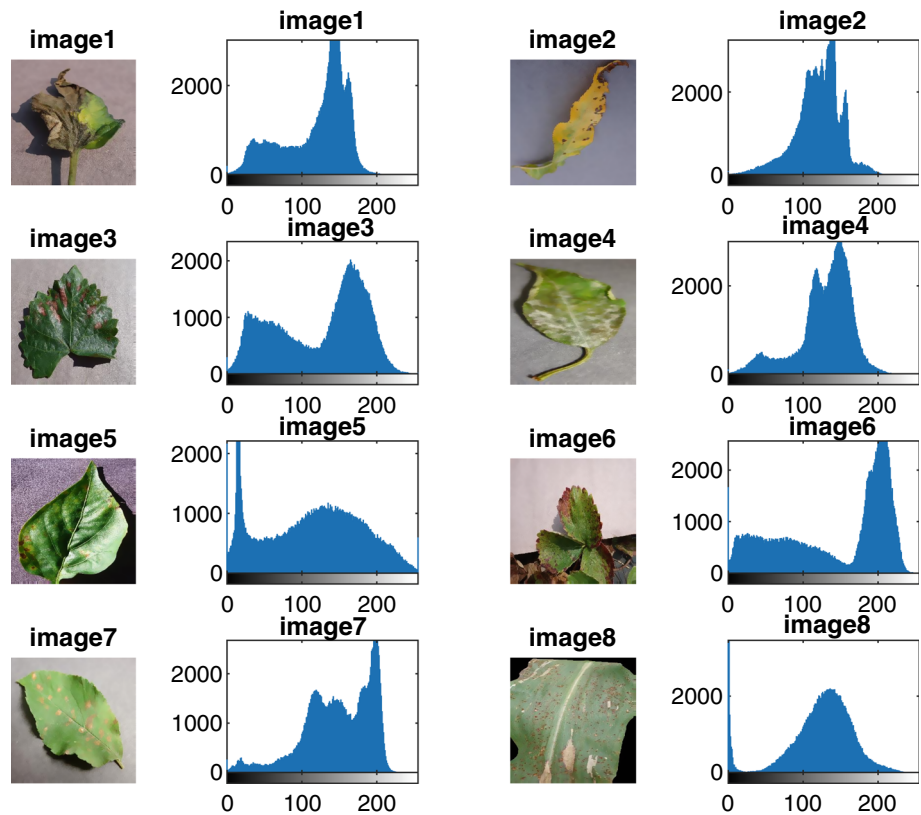


Fig. 2 Comparison between MTLABC and CS in terms of how many experiments, at each level, in which MTLABC or CS achieved the best value of objective function (At each level, there are eight images and for each image there are red, green and blue channels, treated as individual images. So, in total, 24 images exist for each level, each of which is called an experiment)

SSIM and PSNR, the higher the FSIM index is, the better the quality of the segmented image is.

3.4 FSIMc index

The FSIM index is developed to be used with grayscale images or the luminance of color images. The FSIM can be easily extended to be used with color images by incorporating the chrominance information of color images. This extension of the FSIM index is known as FSIMc. Firstly, the RGB color image is transformed into YIQ color space where Y denotes the luminance information, and I and Q contain the chrominance information. This transformation can be obtained by Eq. (20).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (20)$$

We can compare the similarity between the chromatic channels of the original and segmented images as follows:

$$S_I(x) = \frac{2 \cdot I_1(x) \cdot I_2(x) + T_3}{I_1^2(x) + I_2^2(x) + T_3} \quad (21)$$

$$S_Q(x) = \frac{2 \cdot Q_1(x) \cdot Q_2(x) + T_4}{Q_1^2(x) + Q_2^2(x) + T_4} \quad (22)$$

where I_1 and Q_1 are the chromatic channels of the original image while I_2 and Q_2 are the chromatic channels of the segmented image. T_3 and T_4 are, like T_1 and T_2 , positive constant. Since the two chromatic components, I and Q ,

Fig. 3 Convergence curves of the TLBO, TLABC, ABC, CS and MTLABC applied to the first image (red channel) with different numbers of levels ($k = 5, 7, 9, 11, 13, 15, 17, 20$)

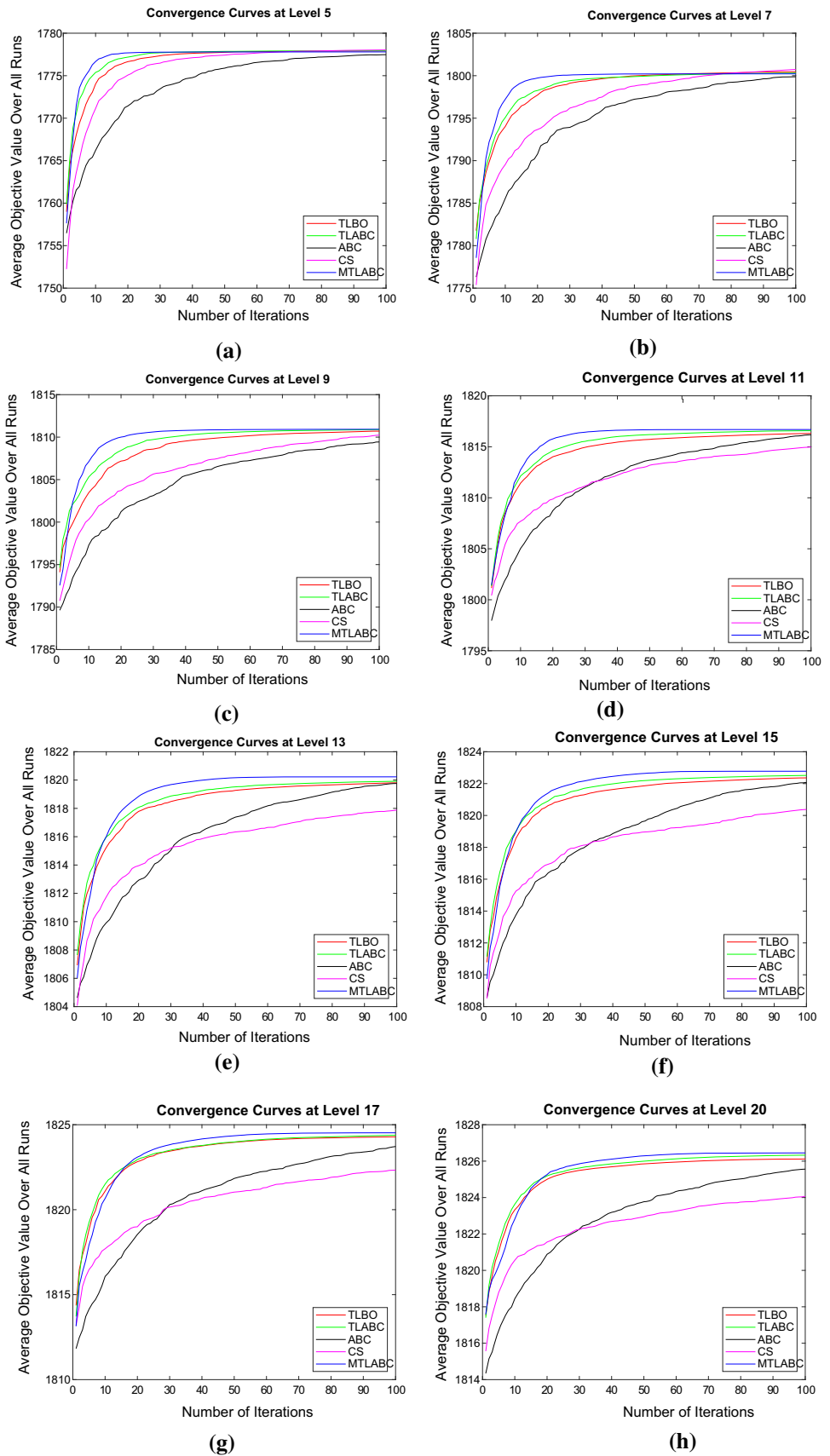


Table 2 Mean CPU time in seconds at level 5 (standard deviation)

Image ID	TLBO	TLABC	ABC	CS	MTLABC
1	3.118 (0.239)	3.708 (0.154)	2.201 (0.182)	3.79 (0.206)	3.687 (0.123)
2	3.013 (0.163)	3.643 (0.125)	1.963 (0.191)	3.886 (0.144)	3.654 (0.130)
3	3.029 (0.154)	3.685 (0.118)	1.509 (0.152)	3.661 (0.133)	3.668 (0.142)
4	3.039 (0.172)	3.666 (0.098)	2.235 (0.247)	3.747 (0.147)	3.671 (0.115)
5	3.027 (0.154)	3.65 (0.141)	1.489 (0.163)	3.659 (0.129)	3.661 (0.139)
6	3.018 (0.169)	3.683 (0.149)	1.892 (0.249)	3.688 (0.148)	3.676 (0.119)
7	3.023 (0.178)	3.686 (0.162)	1.865 (0.200)	3.728 (0.135)	3.662 (0.145)
8	3.016 (0.163)	3.679 (0.128)	1.771 (0.212)	3.667 (0.156)	3.651 (0.134)

have roughly the same dynamic range, T_3 and T_4 can be equal to each other. Then the two similarity components, S_I and S_Q , are combined together to yield a single value according to Eq. (23).

$$S_C(x) = S_I(x) \cdot S_Q(x) \quad (23)$$

Lastly, the FSIMc can be defined as

$$FSIMc = \frac{\sum_{x \in \Omega} S_L(x) \cdot [S_C(x)]^\lambda \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \quad (24)$$

where λ is a positive parameter indicating the significance of the two chromatic components.

4 Experimental results

In this section, we show the results of applying five metaheuristic optimization algorithms to the multilevel segmentation problem. These metaheuristic optimization algorithms are ABC, CS, TLBO, TLABC and MTLABC. The first four are well-known metaheuristic optimization algorithms in the literature, while the last one is a modified version of TLABC proposed in this paper. Eight color images of plant diseases were selected randomly from New Plant Diseases Dataset available on [3] and used as the benchmark to compare the five metaheuristic optimization algorithms. Figure 1 shows these eight color images together with their histograms. As it is shown in Fig. 1, the images are with different levels of complexity. One of these images is unimodal (with one peak), some of them are bimodal (with two peaks) and the others are multimodal. This variety of number of peaks in images histograms makes it a difficult task for the metaheuristic algorithms to find the optimal thresholding values. Each of the five metaheuristic optimization algorithms was repeated 50 times and the averages of these 50 times were used for comparison.

To compare the performance of the five optimization algorithms, five performance criteria were used. These five criteria are the objective function, CPU time, PSNR, SSIM

and FSIMc. The experiments were conducted on MATLAB R2018b installed on a laptop with Intel Core i7-2670QM CPU (2.20GHz) and 6-GB RAM.

4.1 Objective function

As mentioned before, the Otsu's function was used as the objective function of the metaheuristic optimization algorithms with different numbers of thresholds. The objective function was found for each of the three channels of the RGB color images individually. Table 5 shows the values of the objective function for the red channel of the eight images using different numbers of thresholds ($k = 5, 7, 9, 11, 15, 17$ and 20 where k is the number of the thresholds). Likewise, Tables 6 and 7 show the values of the objective function for the green and blue channels. For each image, the best results at every threshold are boldfaced. In Table 5, MTLABC achieved the best results in 43 out of 64 while CS obtained the best results in 21 out of 64. In Table 6, MTLABC obtained the best results in 48 out of 64 while CS achieved the best results in 16 out of 64. In Table 7, MTLABC achieved the best results in 54 out of 64 while CS achieved the best results in 10 out of 64. It is clearly seen that CS was almost the best when the number of thresholds was small while MTLABC was the best when the number of thresholds was large. Figure 2 shows a comparison between the MTLABC and CS which are always the best two algorithms in terms of the objective function. At each level, there are eight color images with red, green and blue channels. In fact, we treat these three channels as individual images so there are 24 image (24 experiments) at each level. From Fig. 2, we see that CS is the best in most of the 24 experiments at low levels ($k = 5$ and 7) while MTLABC is the best in most, and often all, of the 24 experiments at high levels ($k = 9, 11, 13, 15, 17$ and 20). Therefore, MTLABC performs well in high dimensions and as the dimension (number of levels) increases, the MTLABC outperforms the other four algorithms more significantly.

Figure 3 displays the convergence curves of the five algorithms for the red channel of the first image at all the

Table 3 *P* values and Chi-square values of Friedman test of the red, green and blue channels at different levels

Red			Green			Blue		
Level	Chi	<i>p</i> value	Level	Chi	<i>p</i> value	Level	Chi	<i>p</i> value
5	25.90	3.31E−05	5	26.80	2.18E−05	5	24.40	6.64E−05
7	25.30	4.38E−05	7	25.30	4.38E−05	7	24.10	7.63E−05
9	24.10	7.63E−05	9	25.10	4.80E−05	9	25.70	3.64E−05
11	25.00	5.03E−05	11	26.30	2.75E−05	11	28.40	1.03E−05
13	26.30	2.75E−05	13	25.90	3.31E−05	13	25.90	3.31E−05
15	26.30	2.75E−05	15	25.90	3.31E−05	15	26.30	2.75E−05
17	26.30	2.75E−05	17	26.30	2.75E−05	17	26.30	2.75E−05
20	26.90	2.08E−05	20	25.70	3.64E−05	20	25.70	3.64E−05

levels. The convergence curves show how the average values of the objective function change with respect to the number of iterations. Each algorithm was run 50 times so we report in the graph the average values of the objective function. According to Fig. 3, MTLABC achieves the best (fastest) convergence among the five optimization algorithms; in contrast, CS and ABC exhibit the slowest convergence. As a results, the modifications we added to TLABC improves the convergence of the algorithm, not to mention its improved accuracy.

4.2 CPU time

As mentioned before, each one of the five algorithms is repeated 50 times; thus, the mean CPU time in seconds, together with the standard deviation of the 50 repetitions at level 5, as an example, is shown in Table 2. As shown in Table 2, ABC has less mean CPU time in all the eight images since the ABC involves fewer computational as compared to the other algorithms. On the other hand, the computation time of the modified version of TLABC, MTLABC, is comparable to TLABC, although the results and convergence of MTLABC are better than TLABC.

4.3 PSNR

As stated previously, whenever the value of the PSNR index is high, the quality of the segmented image is good and accordingly the performance of the optimization algorithm is high. Table 8 shows the PSNR values for the

five algorithms at the different numbers of the thresholds where the best results were boldfaced. As shown in Table 8, the performance of the MTLABC was the best, in terms of the PSNR index, in 59 out of 64. On the other hand, CS was the best in the remaining five cases. It is also noted that the MTLABC was always the best when the number of the thresholds is high (when $k = 11, 13, 15, 17$ and 20).

4.4 SSIM

Like PSNR, the performance of the optimization algorithm is high when the SSIM index is high. Table 9 shows the SSIM values for the five algorithms at the different numbers of the thresholds. The best results were boldfaced. The MTLABC was the algorithm with the highest SSIM values in 61 out of 64 while CS achieved the best performance in the remaining three cases. As with the PSNR index, the MTLABC was always the best when the number of the thresholds is high (when $k = 9, 11, 13, 15, 17$ and 20).

4.5 FSIM

The extension of the FSIM for the color images was used as other performance metrics. The optimization algorithms with a good performance show high values of the FSIMc index, similar to the PSNR and SSIM indices. Table 10 shows the FSIMc values for the five algorithms at the different numbers of the thresholds. The MTLABC was the algorithm with the best performance in 60 cases out of 64

Table 4 *P* values of the Wilcoxon signed-rank test of the red, green and blue channels

Channels	MTLABC versus ABC	MTLABC versus TLBO	MTLABC versus CS	MTLABC versus TLABC
Red	3.53E−12	3.53E−12	9.13E−08	3.53E−12
Green	3.53E−12	3.53E−12	6.78E−08	3.53E−12
Blue	3.53E−12	3.53E−12	1.05E−10	3.53E−12

Table 5 Averages of the Otsu objective function (red)

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	1747.51	658.80	3726.07	793.66	4013.45	4648.52	1674.75	2693.51
	TLBO	1744.30	660.84	3722.03	793.29	4008.44	4653.33	1676.91	2689.66
	CS	1778.03	697.28	3768.69	834.49	4061.68	4700.12	1715.55	2731.96
	TLABC	1747.94	659.24	3726.01	787.87	4014.36	4654.27	1673.13	2692.97
	MTLABC	1777.80	696.73	3768.54	834.25	4061.49	4699.94	1714.93	2731.87
7	ABC	1772.52	675.59	3771.00	821.40	4073.05	4701.45	1712.26	2727.33
	TLBO	1773.86	681.89	3767.34	821.03	4071.71	4701.80	1712.03	2729.22
	CS	1800.66	712.16	3805.22	854.27	4112.90	4733.11	1743.93	2759.48
	TLABC	1772.67	678.68	3767.67	819.61	4068.95	4700.03	1713.62	2728.34
	MTLABC	1800.43	712.69	3804.72	854.27	4112.74	4732.20	1743.13	2758.50
9	ABC	1786.82	681.33	3793.90	831.44	4101.61	4723.02	1731.06	2746.98
	TLBO	1785.02	675.84	3793.06	834.32	4103.94	4724.07	1729.75	2746.27
	CS	1810.07	717.76	3823.27	862.61	4135.84	4750.31	1755.26	2770.78
	TLABC	1785.02	683.55	3792.46	831.02	4104.91	4723.16	1729.62	2748.59
	MTLABC	1810.94	718.76	3823.26	862.90	4135.62	4750.29	1755.63	2770.49
11	ABC	1795.69	652.61	3810.45	842.97	4119.46	4738.69	1743.59	2758.06
	TLBO	1796.08	658.39	3808.55	843.55	4123.65	4737.25	1742.03	2758.01
	CS	1814.81	721.14	3832.55	867.02	4148.07	4759.42	1762.08	2776.59
	TLABC	1796.61	640.48	3810.89	841.16	4121.24	4738.71	1741.66	2758.90
	MTLABC	1816.58	722.88	3833.77	867.98	4148.63	4759.94	1763.21	2776.55
13	ABC	1800.85	655.08	3819.13	842.70	4134.50	4747.86	1750.82	2766.49
	TLBO	1803.73	589.96	3820.76	844.92	4134.51	4747.82	1750.76	2765.68
	CS	1818.03	723.07	3838.24	869.56	4154.71	4765.13	1766.42	2780.02
	TLABC	1801.52	587.51	3819.89	845.46	4134.72	4748.75	1751.31	2765.73
	MTLABC	1820.25	725.33	3840.37	871.29	4156.44	4766.26	1767.63	2780.93
15	ABC	1807.81	372.41	3826.82	849.52	4142.89	4755.29	1756.11	2771.36
	TLBO	1805.96	530.33	3827.87	817.95	4143.48	4756.27	1756.71	2770.78
	CS	1820.35	724.53	3842.48	871.22	4159.68	4768.88	1769.51	2782.38
	TLABC	1806.79	493.32	3827.37	816.06	4142.91	4755.65	1757.38	2770.53
	MTLABC	1822.71	726.89	3844.82	873.58	4161.79	4770.59	1771.04	2783.65
17	ABC	1808.35	388.38	3832.98	844.79	4149.24	4759.45	1761.13	2774.88
	TLBO	1809.48	288.41	3832.86	853.52	4150.68	4759.99	1760.86	2774.56
	CS	1822.04	725.78	3845.38	872.97	4163.03	4771.64	1771.62	2784.04
	TLABC	1808.92	333.39	3832.79	814.61	4150.03	4759.32	1761.18	2774.34
	MTLABC	1824.51	728.40	3847.98	874.90	4165.55	4773.74	1773.28	2785.67
20	ABC	1813.21	210.73	3838.97	648.63	4157.01	4764.91	1766.06	2778.75
	TLBO	1813.64	170.42	3838.24	751.91	4156.34	4766.13	1765.91	2778.55
	CS	1823.91	726.83	3848.53	874.70	4166.70	4774.61	1774.04	2786.11
	TLABC	1813.44	241.47	3838.97	810.45	4156.64	4765.50	1766.36	2778.86
	MTLABC	1826.31	728.64	3851.19	876.70	4169.49	4776.89	1775.93	2787.78

cases while CS achieved the best performance in the remaining four cases. As with the other metrics, the MTLABC was always the best when the number of the thresholds is high (when $k = 11, 13, 15, 17$ and 20).

4.6 Statistical analysis

Due to their stochastic behavior, statistical tests are needed to prove statistically significant differences between the

Table 6 Averages of the Otsu objective function (green)

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	1361.95	506.82	2591.06	758.53	3715.39	4236.52	1577.97	2887.48
	TLBO	1358.82	502.81	2591.84	755.77	3711.94	4236.28	1573.88	2885.58
	CS	1395.34	545.69	2638.67	793.90	3766.70	4283.56	1614.58	2920.16
	TLABC	1356.43	507.11	2589.42	754.83	3715.09	4236.71	1577.07	2883.48
	MTLABC	1395.20	545.74	2638.54	793.82	3766.55	4283.43	1613.89	2919.99
7	ABC	1385.78	520.20	2638.26	780.04	3771.79	4281.06	1608.14	2918.49
	TLBO	1382.86	515.10	2634.99	779.13	3781.11	4283.49	1607.44	2918.76
	CS	1416.25	557.34	2676.64	814.80	3821.63	4316.89	1637.60	2947.07
	TLABC	1381.08	516.04	2634.47	782.26	3776.39	4282.35	1607.85	2916.99
	MTLABC	1416.02	557.99	2676.27	813.18	3820.99	4316.49	1637.43	2946.85
9	ABC	1396.64	512.05	2662.25	793.87	3809.32	4303.87	1625.10	2935.13
	TLBO	1396.47	507.04	2662.69	794.75	3811.56	4304.57	1625.92	2935.37
	CS	1424.13	562.00	2694.90	822.41	3846.08	4333.57	1649.09	2958.55
	TLABC	1397.98	506.52	2660.24	793.64	3814.18	4306.39	1623.29	2934.83
	MTLABC	1425.65	563.54	2695.08	822.68	3845.13	4334.05	1649.17	2957.70
11	ABC	1407.02	346.24	2677.99	803.63	3831.71	4320.04	1636.59	2946.01
	TLBO	1407.85	409.00	2678.04	804.46	3831.16	4318.51	1636.28	2945.93
	CS	1428.51	564.39	2703.60	826.32	3858.76	4342.44	1654.90	2963.72
	TLABC	1407.07	470.11	2679.05	803.74	3831.41	4319.94	1636.45	2947.32
	MTLABC	1431.13	566.65	2704.61	827.58	3858.78	4343.70	1655.99	2964.10
13	ABC	1415.33	281.65	2689.16	809.51	3844.94	4328.88	1642.96	2953.52
	TLBO	1414.79	275.13	2689.35	811.25	3844.52	4328.18	1643.92	2953.51
	CS	1431.88	565.79	2709.26	829.20	3866.01	4347.77	1658.44	2967.11
	TLABC	1412.82	176.36	2689.32	808.10	3845.78	4329.04	1643.46	2952.84
	MTLABC	1434.30	568.90	2710.71	830.60	3866.55	4349.76	1659.99	2968.07
15	ABC	1418.57	154.35	2698.64	814.83	3854.14	4334.99	1647.37	2959.20
	TLBO	1417.98	222.92	2697.10	813.44	3853.63	4335.72	1649.14	2958.64
	CS	1433.65	567.44	2712.95	831.15	3870.80	4351.00	1660.80	2969.56
	TLABC	1418.15	167.57	2697.43	813.68	3853.20	4336.65	1648.71	2958.58
	MTLABC	1436.76	570.42	2714.94	832.95	3871.90	4353.84	1662.70	2970.67
17	ABC	1421.21	109.76	2702.91	809.57	3861.41	4340.58	1652.19	2961.82
	TLBO	1420.78	97.44	2703.27	799.22	3860.51	4340.88	1652.17	2961.74
	CS	1435.65	568.46	2716.02	832.77	3874.26	4353.58	1662.67	2971.27
	TLABC	1420.19	119.56	2702.18	815.80	3860.71	4340.07	1651.44	2962.54
	MTLABC	1438.35	571.59	2718.14	834.63	3875.92	4356.68	1664.67	2972.58
20	ABC	1423.79	11.15	2708.36	803.27	3866.59	4346.48	1656.60	2966.11
	TLBO	1425.60	11.17	2708.03	762.78	3867.18	4345.95	1656.52	2966.54
	CS	1437.63	544.66	2719.05	834.30	3877.65	4356.54	1664.66	2973.06
	TLABC	1424.75	31.99	2708.69	751.29	3866.47	4346.89	1655.86	2965.98
	MTLABC	1440.16	572.83	2721.14	836.34	3879.76	4359.62	1666.64	2974.64

different metaheuristic optimization algorithms used in this study.

In this paper, the Friedman test and the Wilcoxon signed-rank test, which are two widely used nonparametric hypothesis tests, were used to test whether or not a significant difference between the used optimization algorithms exists. In particular, we want to show that

MTLABC, which is the proposed optimization algorithm in this paper, is statistically different from the other optimization algorithms proposed in the literature and used in this paper.

In the Friedman test, the null hypothesis, H_0 , is that the medians of the optimization algorithms are equal while the alternative hypothesis, H_1 , is that their medians are

Table 7 Averages of the Otsu objective function (blue)

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	1752.27	1205.36	3892.61	1739.80	4628.33	6237.57	2221.20	2146.33
	TLBO	1749.31	1205.94	3887.89	1745.77	4629.25	6235.85	2218.00	2146.57
	CS	1786.50	1257.13	3933.70	1784.69	4676.92	6273.61	2260.00	2188.27
	TLABC	1749.40	1208.67	3887.04	1742.59	4636.56	6232.82	2216.98	2148.00
	MTLABC	1786.30	1257.33	3933.62	1784.72	4676.83	6273.60	2259.97	2188.13
7	ABC	1774.31	1185.10	3936.26	1772.83	4685.72	6272.16	2252.86	2184.55
	TLBO	1772.28	1179.10	3940.31	1772.62	4686.92	6274.41	2259.37	2182.01
	CS	1806.05	1273.23	3973.38	1807.46	4726.50	6301.75	2293.34	2217.74
	TLABC	1775.47	1192.32	3933.95	1773.83	4685.94	6271.29	2254.97	2182.90
	MTLABC	1806.60	1274.25	3973.22	1808.32	4726.24	6301.63	2293.11	2217.74
9	ABC	1788.41	1022.32	3964.91	1788.13	4718.78	6291.30	2276.73	2202.48
	TLBO	1789.92	979.13	3963.04	1789.43	4720.51	6291.21	2272.61	2202.32
	CS	1814.40	1280.56	3992.65	1818.49	4747.89	6315.17	2306.60	2229.01
	TLABC	1787.95	858.88	3963.97	1788.84	4718.87	6291.39	2276.31	2205.03
	MTLABC	1816.70	1282.85	3993.24	1820.56	4748.40	6315.85	2307.35	2229.79
11	ABC	1795.78	540.84	3978.21	1801.02	4736.88	6304.80	2288.16	2214.20
	TLBO	1795.72	691.43	3979.18	1800.53	4736.30	6303.78	2285.27	2214.55
	CS	1819.37	1283.85	4002.80	1823.81	4760.43	6322.80	2313.49	2235.01
	TLABC	1796.62	693.50	3979.55	1801.83	4737.49	6303.14	2291.42	2214.59
	MTLABC	1822.70	1287.18	4004.87	1827.57	4762.87	6324.32	2315.48	2236.36
13	ABC	1802.85	322.79	3990.03	1807.26	4750.04	6311.56	2295.45	2222.47
	TLBO	1801.43	252.08	3991.77	1808.14	4750.30	6311.18	2296.69	2221.76
	CS	1822.33	1286.00	4009.40	1828.37	4768.81	6327.04	2317.60	2238.51
	TLABC	1802.09	422.67	3990.53	1807.88	4749.20	6311.05	2294.70	2222.34
	MTLABC	1826.50	1290.29	4011.91	1831.63	4771.51	6329.53	2319.94	2240.71
15	ABC	1807.59	226.41	3998.97	1811.58	4756.98	6316.96	2302.20	2227.21
	TLBO	1805.90	51.03	3998.14	1810.86	4757.29	6316.76	2299.27	2227.94
	CS	1824.74	1286.98	4013.75	1831.17	4773.80	6329.97	2320.59	2241.15
	TLABC	1806.68	250.15	3997.91	1810.43	4759.02	6316.70	2299.25	2227.49
	MTLABC	1828.99	1291.90	4016.60	1834.48	4776.90	6332.79	2323.73	2243.27
17	ABC	1809.18	25.67	4003.96	1815.27	4764.36	6320.11	2214.99	2232.56
	TLBO	1809.85	152.81	4004.03	1814.76	4763.39	6320.22	2258.07	2231.67
	CS	1826.75	1285.68	4016.84	1832.72	4777.56	6332.43	2323.04	2242.95
	TLABC	1739.30	76.02	4004.38	1815.83	4764.85	6320.20	2303.87	2231.45
	MTLABC	1830.68	1292.72	4019.96	1836.62	4780.51	6335.29	2325.91	2245.38
20	ABC	1593.53	25.66	4010.18	1819.98	4771.51	6325.65	2311.24	2235.45
	TLBO	1704.95	25.60	4009.58	1743.87	4771.57	6325.24	2213.11	2235.61
	CS	1828.71	874.55	4020.41	1835.27	4781.52	6334.75	2325.50	2245.08
	TLABC	1703.93	50.30	4010.70	1820.50	4771.12	6324.93	2169.78	2235.90
	MTLABC	1832.39	1293.74	4023.28	1838.70	4784.25	6337.58	2328.34	2247.27

different. Therefore, rejecting the null hypothesis in favor of the alternative demonstrates that the distinct optimization algorithms are different from each other. In addition to hypotheses, the significance level of the test, α , is defined as the probability of rejecting the null hypothesis while it is, in fact, true. Accordingly, as the significant level goes down, it becomes more difficult to reject the null

hypothesis. Thus, the smallest significance level at which we still can reject the null hypothesis is an important factor known as the p value. The p value indicates the level of confidence when rejecting the null the hypothesis. The smaller the p value is, the more confident we reject the null hypothesis. Moreover, when the p value for some sample is

Table 8 PSNR values of ABC, TLBO, CS, TLABC and MTLABC

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	21.4519	20.6826	19.7454	21.0351	19.8875	19.5585	20.7078	20.1612
	TLBO	23.4913	23.1349	21.5271	23.7153	21.2922	21.1640	22.7826	22.4223
	CS	21.3993	20.6630	19.8801	21.0682	19.8020	19.6381	20.5004	20.3804
	TLABC	21.3662	21.0854	19.7406	20.9413	19.9486	19.8307	20.4650	20.2629
	MTLABC	23.6960	24.1549	21.5741	23.7430	21.2913	21.1092	22.8811	22.5227
7	ABC	23.1229	22.1513	21.7013	22.8770	21.6660	21.7989	22.4709	22.3639
	TLBO	25.7374	26.6016	23.8553	25.5849	23.6378	24.0100	25.6298	25.3456
	CS	22.9558	22.5667	21.6506	22.7729	21.7452	21.7914	22.6207	22.1294
	TLABC	23.0449	22.0943	21.6239	22.7684	21.6952	21.8142	22.5572	22.7046
	MTLABC	26.1455	27.4517	24.1029	26.0546	23.6375	23.9471	25.6797	25.7734
9	ABC	24.3756	22.6977	23.2720	23.9381	23.1338	23.0835	24.0430	24.2048
	TLBO	27.1116	28.2804	25.6515	26.9949	25.2735	26.0804	27.4350	27.4592
	CS	24.5149	22.3388	23.2459	24.0733	23.2287	23.2321	23.9555	24.2146
	TLABC	24.2990	21.9082	23.2930	24.0074	23.2886	23.4169	24.0146	24.5382
	MTLABC	28.1404	29.5918	26.1124	27.8603	25.3827	26.0745	27.6536	28.3691
11	ABC	25.3261	21.4343	24.3947	25.2346	24.4193	24.6869	25.2928	25.7871
	TLBO	28.1361	29.2825	27.0067	27.9346	26.5839	27.5754	28.7102	28.9694
	CS	25.3175	22.5466	24.5036	25.2004	24.4539	24.5941	25.1587	25.7176
	TLABC	25.4675	22.2988	24.6764	25.1219	24.4284	24.7090	25.3415	25.7440
	MTLABC	29.9982	31.2799	27.8259	29.6387	26.8517	28.0278	29.2385	29.9912
13	ABC	26.3770	22.9932	25.6100	25.8623	25.4925	25.8175	26.2801	27.0722
	TLBO	28.9547	30.1277	28.0310	28.9297	27.6055	28.8920	29.6576	30.2207
	CS	26.3268	22.7026	25.5899	25.9451	25.5082	25.8033	26.3664	27.0624
	TLABC	26.1508	22.8730	25.5728	25.9476	25.4798	25.8204	26.2128	27.0477
	MTLABC	31.4723	32.5600	29.1929	31.1482	28.1240	29.4302	30.4385	31.2835
15	ABC	27.2286	22.8586	26.6585	26.6541	26.3612	26.6210	27.0638	28.0834
	TLBO	29.9207	30.6809	29.0033	29.7107	28.4775	29.6922	30.4587	31.3258
	CS	26.9929	23.1184	26.5758	26.5181	26.3523	26.8198	27.1313	28.0345
	TLABC	27.0275	23.4152	26.5882	26.5818	26.3599	26.7518	27.0798	27.9278
	MTLABC	32.5962	33.5383	30.3721	32.3773	29.3655	30.6781	31.5693	32.3451
17	ABC	27.5057	23.8960	27.3017	26.5529	27.2309	27.4079	27.8978	29.0520
	TLBO	30.5057	30.9785	29.9242	30.3560	29.3209	30.5441	31.2501	32.0715
	CS	27.5591	23.6777	27.4189	27.1371	27.2308	27.5173	27.8168	28.7147
	TLABC	27.6286	23.5240	27.3488	26.9419	27.2338	27.4492	27.7855	28.9499
	MTLABC	33.5843	34.3395	31.3955	33.3206	30.4045	31.7632	32.5085	33.2783
20	ABC	27.8910	24.4012	28.4517	27.6364	28.3187	28.6648	28.9606	30.0368
	TLBO	31.6857	30.1578	30.9902	31.4208	30.3588	31.5969	32.1973	33.1929
	CS	28.4037	24.7893	28.4651	27.0071	28.3020	28.7410	28.6187	30.0771
	TLABC	28.3340	24.6921	28.5661	27.1604	28.2801	28.7129	28.6542	30.0065
	MTLABC	34.7605	35.0448	32.6470	34.5037	31.7799	33.1081	33.7642	34.5181

less than the significant level of the test, the null hypothesis gets rejected by that test.

We applied the Friedman test to the three channels of the RGB color images individually. In other words, the red, green and blue channels of any image are treated as three different images, and Table 3 shows the *p* values along with the Chi-square χ^2 values of the Friedman test at

different threshold numbers. From a χ^2 -distribution, the 0.95 quantile (the critical value at significant level 0.05) of χ^2 -distribution with (5-1) degrees of freedom is 9.488 and all the χ^2 values in Table 4 are significantly greater than this critical value. That means the null hypothesis is confidently rejected and the optimization algorithms are proved different. Likewise, the very small *p* values mean

Table 9 SSIM values of ABC, TLBO, CS, TLABC and MTLABC

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	0.7670	0.6814	0.6990	0.7272	0.7840	0.6181	0.6901	0.7782
	TLBO	0.8439	0.8400	0.7837	0.8603	0.8149	0.7610	0.7696	0.8565
	CS	0.7715	0.6783	0.7043	0.7222	0.7797	0.6299	0.6846	0.7838
	TLABC	0.7731	0.6977	0.6992	0.7185	0.7852	0.6355	0.6799	0.7789
	MTLABC	0.8495	0.8608	0.7852	0.8594	0.8154	0.7607	0.7768	0.8578
7	ABC	0.8261	0.6999	0.7810	0.7840	0.8276	0.7206	0.7420	0.8405
	TLBO	0.8922	0.8991	0.8604	0.8902	0.8534	0.8217	0.8500	0.9054
	CS	0.8191	0.7221	0.7808	0.7868	0.8267	0.7245	0.7518	0.8359
	TLABC	0.8194	0.7128	0.7798	0.7813	0.8291	0.7301	0.7502	0.8477
	MTLABC	0.8972	0.9156	0.8664	0.8968	0.8564	0.7921	0.8540	0.9115
9	ABC	0.8527	0.7122	0.8340	0.8122	0.8611	0.7653	0.7889	0.8789
	TLBO	0.9122	0.9188	0.8995	0.9122	0.8743	0.8593	0.8802	0.9287
	CS	0.8533	0.6855	0.8326	0.8189	0.8599	0.7651	0.7886	0.8783
	TLABC	0.8484	0.6373	0.8360	0.8160	0.8635	0.7742	0.7872	0.8856
	MTLABC	0.9258	0.9398	0.9092	0.9222	0.8902	0.8645	0.8861	0.9392
11	ABC	0.8715	0.6141	0.8632	0.8497	0.8885	0.8161	0.8180	0.9061
	TLBO	0.9248	0.9307	0.9227	0.9249	0.9009	0.8923	0.9000	0.9408
	CS	0.8712	0.6744	0.8654	0.8425	0.8859	0.8151	0.8194	0.9054
	TLABC	0.8753	0.6361	0.8723	0.8444	0.8848	0.8158	0.8196	0.9047
	MTLABC	0.9481	0.9540	0.9372	0.9412	0.9169	0.9052	0.9088	0.9530
13	ABC	0.8909	0.7010	0.8913	0.8551	0.8987	0.8455	0.8444	0.9216
	TLBO	0.9325	0.9397	0.9355	0.9353	0.9146	0.9186	0.9133	0.9493
	CS	0.8886	0.7084	0.8890	0.8575	0.9023	0.8467	0.8453	0.9227
	TLABC	0.8879	0.6899	0.8894	0.8577	0.8976	0.8481	0.8469	0.9223
	MTLABC	0.9630	0.9642	0.9536	0.9551	0.9339	0.9289	0.9250	0.9624
15	ABC	0.9045	0.7154	0.9099	0.8708	0.9155	0.8681	0.8613	0.9329
	TLBO	0.9423	0.9442	0.9474	0.9423	0.9231	0.9275	0.9240	0.9555
	CS	0.9000	0.6826	0.9079	0.8690	0.9108	0.8738	0.8623	0.9316
	TLABC	0.9013	0.7296	0.9096	0.8644	0.9121	0.8684	0.8584	0.9308
	MTLABC	0.9711	0.9696	0.9650	0.9647	0.9551	0.9454	0.9398	0.9699
17	ABC	0.9080	0.7109	0.9196	0.8552	0.9257	0.8826	0.8752	0.9415
	TLBO	0.9460	0.9410	0.9560	0.9488	0.9344	0.9406	0.9340	0.9592
	CS	0.9083	0.7378	0.9224	0.8751	0.9265	0.8877	0.8752	0.9381
	TLABC	0.9094	0.7046	0.9206	0.8686	0.9240	0.8837	0.8745	0.9397
	MTLABC	0.9776	0.9723	0.9723	0.9707	0.9650	0.9567	0.9504	0.9747
20	ABC	0.9083	0.7662	0.9360	0.8778	0.9394	0.9066	0.8948	0.9501
	TLBO	0.9563	0.8919	0.9645	0.9566	0.9442	0.9514	0.9430	0.9654
	CS	0.9203	0.7744	0.9371	0.8676	0.9374	0.9092	0.8849	0.9506
	TLABC	0.9190	0.7733	0.9374	0.8630	0.9391	0.9089	0.8866	0.9483
	MTLABC	0.9826	0.9735	0.9787	0.9764	0.9749	0.9676	0.9617	0.9800

that we can reject the null hypothesis with high confidence. As a result, we can conclude that the five P-metaheuristic algorithms are not the same according to the Friedman test (indicating that the proposed MTLABC might be different from the other four and a follow-up pairwise test is needed to prove that).

The Wilcoxon signed-rank test was used as a follow-up pairwise test, to statistically validate the results and show a significant difference between the proposed MTLABC algorithm and the other optimization algorithms from which MTLABC is a modified version. The Wilcoxon signed-rank test performed a pairwise test between MTLABC and the other four optimization algorithms used

Table 10 FSIMc values of ABC, TLBO, CS, TLABC and MTLABC

Level	Algorithm	Image ID							
		1	2	3	4	5	6	7	8
5	ABC	0.8891	0.8303	0.8948	0.8446	0.9012	0.8660	0.8383	0.8767
	TLBO	0.9202	0.8639	0.9249	0.8813	0.9159	0.8792	0.8711	0.9005
	CS	0.8895	0.8232	0.9032	0.8432	0.8970	0.8644	0.8351	0.8842
	TLABC	0.8880	0.8367	0.8966	0.8408	0.9012	0.8695	0.8342	0.8770
	MTLABC	0.9220	0.8791	0.9258	0.8827	0.9159	0.8804	0.8708	0.9007
7	ABC	0.9121	0.8571	0.9245	0.8772	0.9236	0.8999	0.8751	0.9159
	TLBO	0.9378	0.9178	0.9509	0.9220	0.9414	0.9205	0.9096	0.9386
	CS	0.9095	0.8608	0.9237	0.8792	0.9234	0.9013	0.8728	0.9121
	TLABC	0.9101	0.8552	0.9272	0.8782	0.9236	0.9009	0.8744	0.9163
	MTLABC	0.9435	0.9275	0.9545	0.9235	0.9423	0.9273	0.9106	0.9373
9	ABC	0.9277	0.8697	0.9437	0.8967	0.9402	0.9200	0.8967	0.9373
	TLBO	0.9502	0.9380	0.9626	0.9403	0.9563	0.9472	0.9359	0.9596
	CS	0.9298	0.8666	0.9434	0.8993	0.9407	0.9212	0.9010	0.9371
	TLABC	0.9257	0.8609	0.9433	0.9005	0.9409	0.9251	0.8975	0.9375
	MTLABC	0.9566	0.9506	0.9664	0.9436	0.9592	0.9474	0.9397	0.9596
11	ABC	0.9382	0.8587	0.9512	0.9139	0.9520	0.9381	0.9173	0.9525
	TLBO	0.9568	0.9468	0.9688	0.9492	0.9649	0.9604	0.9522	0.9699
	CS	0.9384	0.8774	0.9535	0.9186	0.9517	0.9357	0.9162	0.9501
	TLABC	0.9373	0.8735	0.9546	0.9135	0.9515	0.9382	0.9160	0.9501
	MTLABC	0.9686	0.9647	0.9746	0.9565	0.9672	0.9648	0.9572	0.9732
13	ABC	0.9459	0.8800	0.9600	0.9245	0.9584	0.9482	0.9298	0.9622
	TLBO	0.9616	0.9536	0.9726	0.9547	0.9692	0.9689	0.9601	0.9759
	CS	0.9482	0.8770	0.9603	0.9253	0.9577	0.9460	0.9317	0.9625
	TLABC	0.9429	0.8748	0.9599	0.9234	0.9582	0.9481	0.9305	0.9616
	MTLABC	0.9793	0.9726	0.9808	0.9695	0.9717	0.9741	0.9678	0.9819
15	ABC	0.9543	0.8768	0.9672	0.9339	0.9642	0.9545	0.9386	0.9700
	TLBO	0.9668	0.9589	0.9759	0.9592	0.9722	0.9732	0.9656	0.9803
	CS	0.9522	0.8855	0.9668	0.9329	0.9638	0.9547	0.9403	0.9672
	TLABC	0.9526	0.8825	0.9669	0.9341	0.9635	0.9564	0.9410	0.9660
	MTLABC	0.9844	0.9780	0.9855	0.9793	0.9779	0.9805	0.9746	0.9869
17	ABC	0.9573	0.8919	0.9693	0.9334	0.9693	0.9595	0.9458	0.9728
	TLBO	0.9697	0.9632	0.9796	0.9633	0.9754	0.9774	0.9705	0.9826
	CS	0.9561	0.8905	0.9714	0.9392	0.9691	0.9609	0.9495	0.9722
	TLABC	0.9557	0.8905	0.9712	0.9360	0.9690	0.9591	0.9470	0.9729
	MTLABC	0.9880	0.9822	0.9889	0.9846	0.9818	0.9841	0.9799	0.9899
20	ABC	0.9626	0.8953	0.9748	0.9446	0.9740	0.9685	0.9566	0.9790
	TLBO	0.9744	0.9601	0.9823	0.9679	0.9782	0.9815	0.9754	0.9869
	CS	0.9640	0.8970	0.9759	0.9369	0.9731	0.9687	0.9565	0.9784
	TLABC	0.9626	0.8981	0.9761	0.9390	0.9733	0.9692	0.9561	0.9792
	MTLABC	0.9912	0.9855	0.9920	0.9891	0.9871	0.9886	0.9852	0.9929

in this paper. If the p value of this test is less than 0.05, then there, indeed, exists a statistically significant difference between the two algorithms which the test compared. Otherwise, the test fails to prove any significant difference between the two algorithms. Table 4 shows the p values of

comparing MTLABC against TLABC, TLBO, ABC and CS using the Wilcoxon signed-rank test. It is obvious that the p values are much smaller than 0.05, such that the significant difference between MTLABC and the other algorithms are demonstrated.

5 Discussion and conclusion

Thresholding is a widely used algorithm for image segmentation owing to its robustness, simplicity and efficiency. In essence, thresholding is deemed an optimization problem. Some objective functions attempt to find the optimal threshold values, efficiently separating the distinct objects in an image. Indeed, Otsu's function is one of the commonly used objective functions of this optimization problem. Like any optimization problem, multiple choices exist to solve the problem and the best option is the one that best suits the problem in hand. In general, there are two broad types of optimization algorithms: deterministic and stochastic optimization algorithms. The latter is used whenever the problem is complicated, like our case, and the deterministic optimization algorithms fail to solve the problem or take unreasonably much time. In this work, we used five different metaheuristic optimization algorithms to solve the multilevel thresholding segmentation problem where Otsu's function was used as the objective function. These five stochastic algorithms are ABC, CS, TLBO, TLABC and a modified version of TLABC proposed in this paper. The benchmark used in this paper consists of eight color images of plant leaves with some kind of plant disease. Normally, a plant disease image comprises three to five distinct parts.

However, we tried multiple numbers of levels and reported the five algorithms' results at each level. The objective function values and three commonly used image quality assessment metrics are used to evaluate and compare the five stochastic performance algorithms. Our proposed algorithm showed the best results in most images and most of the levels in terms of these four performance measures. In fact, MTLABC was the best in all the eight images when the number of levels is high indicating that MTLABC works considerably well in high dimension; CS ranked second after MTLABC. Two nonparametric statistical tests were used to analyze the results statistically. These statistical tests proved that MTLABC, the algorithm proposed in this work, is statistically different from the other four stochastic algorithms: ABC, TLBO, TLABC and CS. The p values obtained in both tests are less than 10^{-4} .

For future study, the data set can be extended into more than eight images, and further stages of image processing, such as feature extraction and classification, can be carried out. The new hybrid stochastic algorithm proposed in this work will be applied to other applications and compared with the already existing state-of-the-art stochastic algorithms in future work.

Conflict of 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

1. Ameer M, Habba M, Jabrane Y (2019) A comparative study of nature inspired optimization algorithms on multilevel thresholding image segmentation. *Multimed Tools Appl* 78:34353–34372
2. Bashish DA, Braik M, Bani-Ahmad S (2011) Detection and classification of leaf diseases using k-means-based segmentation and neural-networks-based classification. *Inf Technol J* 10(2):267–275
3. Bhattarai S. Dataset. <https://www.kaggle.com/vipooool/new-plant-diseases-dataset>. Accessed 1 Dec 2019
4. Chakraborty R, Sushil R, Garg M (2019) An improved PSO-based multilevel image segmentation technique using minimum cross-entropy thresholding. *Arab J Sci Eng* 44(4):3005–3020
5. Chen X, Xu B (2018) Teaching-learning-based artificial bee colony. In: Tan Y, Shi Y, Tang Q (eds) *Advances in swarm intelligence—9th international conference, ICSI 2018, Shanghai, China, June 17–22, 2018, Proceedings, Part I*, vol 10941. Lecture notes in computer science. Springer, pp 166–178
6. Elaziz MA, Bhattacharyya S, Lu S (2019) Swarm selection method for multilevel thresholding image segmentation. *Expert Syst Appl* 138:1–24
7. Ewees AA, Elaziz MA, Al-Qaness MAA, Khalil HA, Kim S (2020) Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation. *IEEE Access* 8:26304–26315. <https://doi.org/10.1109/ACCESS.2020.2971249>
8. Gao H, Pun CM, Kwong S (2016) An efficient image segmentation method based on a hybrid particle swarm algorithm with learning strategy. *Inf Sci* 369:500–521
9. Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. *J R Stat Soc Ser C (Appl Stat)* 28(1):100–108
10. He JH (2020) A new proof of the dual optimization problem and its application to the optimal material distribution of sic/graphene composite. *Rep Mech Eng* 1(1):187–191
11. He L, Huang S (2020) An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. *Appl Soft Comput* 89:106063
12. Hore A, Ziou D (2010) Image quality metrics: Psnr vs. ssim. In: *20th international conference on pattern recognition*
13. Houssein EH, Helmy BE, Oliva D, Elgar AA, Shaban H (2020) A novel black widow optimization algorithm for multilevel thresholding image segmentation. *Expert Syst Appl* 167:114159
14. Jaware TH, Badgajar RD, Patil PG (2012) Crop disease detection using image segmentation. *World J Sci Technol* 2(4):190–194
15. Khan WA (2021) Numerical simulation of Chun-Hui He's iteration method with applications in engineering. *Int J Numer Meth Heat Fluid Flow*. <https://doi.org/10.1108/HFF-04-2021-0245>
16. Kong D, Chang T, Dai W, Wang Q, Sun H (2018) An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy. *Inf Sci* 442–443:54–71
17. Luessi M, Eichmann M, Schuster GM, Katsaggelos AK (2006) New results on efficient optimal multilevel image thresholding. In: *2006 IEEE international conference on image processing, ICIP 2006-proceedings, proceedings-international conference on image processing. ICIP*, pp 773–776
18. Masi M (2005) A step beyond Tsallis and Rényi entropies. *Phys Lett A* 338(3–5):217–224

Compliance with ethical standards

19. Merzban MH, Elbayoumi M (2019) Efficient solution of Otsu multilevel image thresholding: a comparative study. *Expert Syst Appl* 116:299–309
20. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms. *Evol Intel* 10:45–75
21. Oliva D, Elaziz MA, Hinojosa S (2019) Multilevel thresholding for image segmentation based on metaheuristic algorithms. *Studies in computational intelligence*. Springer, Berlin, pp 59–69
22. Rao RV, Savsani VJ, Balic J (2012) Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. *Eng Optim* 44(12):1447–1462
23. Saleh RAA, Akay R (2019) Classification of melanoma images using modified teaching learning based artificial bee colony. In: *European journal of science and technology*. Springer, pp 225–232
24. Shubham S, Bhandari AK (2018) A generalized Masi entropy based efficient multilevel thresholding method for color image segmentation. *Multimed Tools Appl* 78:17197–17238
25. Singh V, Misra AK (2017) Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf Process Agric* 4(1):41–49
26. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
27. Weizheng S, Yachun W, Zhanliang C, Hongda W (2008) Grading method of leaf spot disease based on image processing. In: *2008 international conference on computer science and software engineering*, vol 6, pp 491–494
28. Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: *World Congress on nature and biologically inspired computing (NaBIC)*, pp 210–214 . <https://doi.org/10.1109/NABIC.2009.5393690>
29. Zhang L, Zhang L, Mou X, Zhang D (2011) Fsim: a feature similarity index for image quality assessment. *IEEE Trans Image Process* 20(8):2378–2386
30. Zhang S, Wang H, Huang W, You Z (2018) Plant diseased leaf segmentation and recognition by fusion of superpixel, k-means and phog. *Optik* 157:866–872

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.